

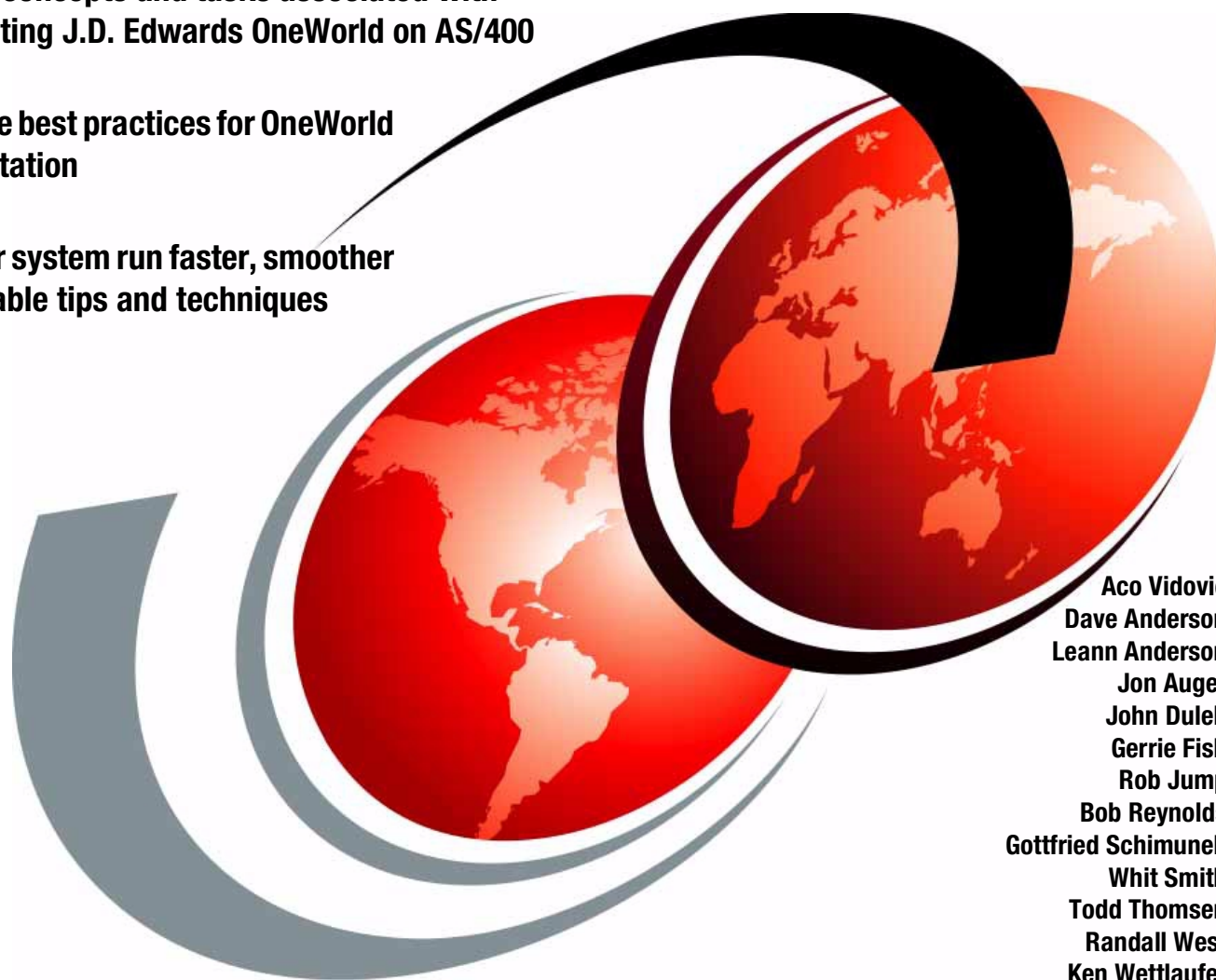
J.D. Edwards OneWorld

Implementation for AS/400

Learn the concepts and tasks associated with implementing J.D. Edwards OneWorld on AS/400

Employ the best practices for OneWorld implementation

Make your system run faster, smoother with valuable tips and techniques



Aco Vidovic
Dave Anderson
Leann Anderson
Jon Auger
John Dulek
Gerrie Fisk
Rob Jump
Bob Reynolds
Gottfried Schimunek
Whit Smith
Todd Thomsen
Randall West
Ken Wettlaufer



International Technical Support Organization

SG24-5195-01

J.D. Edwards OneWorld Implementation for AS/400

July 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix G, "Special notices" on page 849.

Second Edition (July 2000)

This edition applies to V4R4 and V4R5 of OS/400, Program Number 5769-SS1 and primarily to the release B733.2 of OneWorld.

This edition was created or updated on July 10, 2000.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1999, 2000. All rights reserved

Note to U.S Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	xiii
The team that wrote this redbook	xiii
Comments welcome	xviii

Part 1. Understanding the products 1

Chapter 1. Introduction to J.D. Edwards	3
1.1 Company	3
1.2 Products	3
1.2.1 WorldSoftware	4
1.2.2 OneWorld	4
Chapter 2. Introduction to IBM AS/400	7
2.1 AS/400 architecture	9
2.1.1 High level machine interface	9
2.1.2 Hierarchy of microprocessors	10
2.1.3 Object-based architecture	11
2.1.4 Single level storage	11
2.2 64-bit technology	13
2.2.1 Copper chip technology	13
2.2.2 Silicon on Insulator (SOI)	14
2.3 Hot-plugging	14
2.4 High speed link (HSL)	14
2.5 Integrated relational database (DB2 UDB for AS/400)	14
2.6 Data management	15
2.7 Security, user profiles, and authority management	16
2.8 Integrated file system (IFS)	17
2.9 Logical Partitioning (LPAR)	18
2.9.1 LPAR scenarios	18
2.10 Clustering	22
2.11 System management	22
2.11.1 Electronic Customer Support (ECS)	23
2.11.2 Management Central	24
2.11.3 System management facilities	24
2.12 AS/400 advanced technologies	25
2.12.1 Java	25
2.12.2 Web server for e-business	26
2.12.3 Lotus Domino	27
2.12.4 Business intelligence solutions	27
2.13 Why OneWorld on AS/400 system	29
Chapter 3. OneWorld internal architecture	31
3.1 OneWorld tool set	31
3.1.1 OneWorld design tools	31
3.1.2 OneWorld runtime engines	31
3.1.3 OneWorld subsystems	32
3.1.4 Update process for OneWorld tools	33
3.2 OneWorld application code creation and storage	33
3.2.1 OneWorld application specifications: Building blocks	33
3.2.2 Specification storing: Central objects	34
3.2.3 Events and Event Rules	36

3.2.4	OneWorld business functions	37
3.2.5	Update process for applications	37
3.3	OneWorld application development flow	37
3.3.1	OneWorld standard environments and path codes	37
3.3.2	OneWorld objects	38
3.3.3	Client and server packages	43
Chapter 4.	Configurable Network Computing (CNC)	47
4.1	Definition	47
4.1.1	Benefits	47
4.1.2	Recommendations	49
4.2	Environments	51
4.2.1	Environment tables	51
4.3	Path codes	52
4.3.1	At installation	53
4.3.2	At runtime	53
4.3.3	At development	53
4.3.4	Locations	54
4.4	Object Configuration Manager (OCM)	54
4.4.1	Location types	55
4.4.2	Retrieving application data	56
4.5	Data sources	57
4.5.1	Defining data sources	57
4.5.2	Database data sources	57
4.5.3	Logic data sources	58
4.5.4	Required logic and database data sources	58
4.6	Object storage	63
4.7	Object deployment	64
4.7.1	Initial installation	64
4.7.2	Workstation installation	64
4.7.3	Application installation (update packages)	64
4.7.4	JITI	65
4.7.5	Server packages	65
4.8	Middleware	65
4.8.1	JDENet	66
4.8.2	JDEBase	68
Chapter 5.	OneWorld implementation designs	71
5.1	Terminology	71
5.1.1	Different types of servers	71
5.1.2	Distributed processing and distributed data	72
5.2	Implementation design examples	73
5.2.1	Heavy client	73
5.2.2	Lite client	80
5.2.3	Summary	82
Chapter 6.	Database	83
6.1	DB2 UDB for AS/400	83
6.1.1	Data files	84
6.2	AS/400 SQL concepts	86
6.2.1	Terminology	87
6.2.2	Types of SQL statements	89
6.3	DB2 UDB for AS/400 catalog views	90
6.4	DB2 UDB for AS/400 query optimization	91

6.4.1	General optimization strategy	92
6.4.2	Optimization tools	93
6.4.3	Encoded vector index (EVI)	93
6.5	Journaling	94
6.5.1	Introduction to remote journal function	95
6.6	J.D. Edwards database implementation	98
6.6.1	J.D. Edwards database naming conventions	98
6.6.2	Coexistence versus OneWorld only	101
6.6.3	Data sources	101
6.6.4	Journaling and commitment control	102
6.6.5	J.D. Edwards methodology	102
Chapter 7. Work management		109
7.1	AS/400 work management concepts	109
7.1.1	AS/400 system values	109
7.1.2	AS/400 network attributes	109
7.1.3	Subsystems	110
7.1.4	Memory management	116
7.1.5	Jobs	116
7.1.6	Work Control Block Table (WCBT)	119
7.2	Client/server work management	120
7.2.1	Optimized host servers	121
7.2.2	Establishing client/server communications	121
7.2.3	Client/server subsystems	125
7.2.4	Prestart jobs	125
7.2.5	Open Database Connectivity (ODBC) connections	130
7.3	OneWorld work management	130
7.3.1	OneWorld middleware	130
7.3.2	OneWorld process-based design	130
7.3.3	OneWorld AS/400 architecture and process flow	133

Part 2. Implementation137

Chapter 8. Pre-installation and planning		139
8.1	OneWorld customer preparation	140
8.2	Hardware requirements	140
8.2.1	Deployment Server	141
8.2.2	AS/400 Enterprise Server	141
8.2.3	Client workstations	142
8.2.4	Windows Terminal Server	142
8.2.5	Java Application Server	142
8.3	Software requirements	142
8.3.1	Deployment Server	142
8.3.2	AS/400 Enterprise Server	143
8.3.3	Client workstations	143
8.4	Network assessment	144
8.5	Disk space requirements	144
8.6	Environment Checker	145
8.7	Details worksheet	145
8.8	Planning for coexistence with WorldSoftware	145
8.8.1	Coexistent system data source considerations	145
8.8.2	Coexistence control tables	146
8.8.3	Coexistence data dictionary	146

8.9	Central objects storage considerations	146
8.10	Data source planning	146
8.11	AS/400 server preparation	147
8.11.1	Configuring TCP/IP	147
8.11.2	Verifying OS/400 security	147
8.11.3	Creating user ASP for OneWorld journals	147
8.12	OneWorld planning check list	147
Chapter 9. OneWorld installation and maintenance		151
9.1	First time installation	151
9.1.1	Installing the Deployment Server	152
9.1.2	Installing the OneWorld databases	183
9.1.3	Planning and configuring OneWorld servers	184
9.1.4	Enterprise Server installation	221
9.1.5	Configuring and installing client workstations	236
9.1.6	Completing the installation	240
9.2	OneWorld software maintenance	243
9.2.1	Software Application Requests (SARs)	244
9.2.2	Electronic Software Updates (ESUs)	244
9.2.3	OneWorld service packs	244
9.2.4	Application Software Updates (ASUs)	245
9.2.5	Update process	245
9.2.6	Upgrade process	246
Chapter 10. WorldSoftware and OneWorld		247
10.1	Coexistence of WorldSoftware and OneWorld	247
10.1.1	Coexistence and cohabitation	247
10.1.2	Why install OneWorld	247
10.1.3	Structural differences between WorldSoftware and OneWorld	248
10.1.4	Advantages and disadvantages of coexistence	249
10.1.5	Required software levels	249
10.1.6	User profile maintenance	250
10.1.7	Installation issues	250
10.1.8	Data Source setup issues	251
10.1.9	WorldSoftware versus OneWorld tables and indexes	253
10.1.10	Control table comparison	254
10.1.11	Data dictionary file synchronization	254
10.1.12	User Defined Code synchronization	255
10.1.13	Coexistence triggers	255
10.1.14	Non-coexistence applications	256
10.2	Moving from WorldSoftware to OneWorld	256
10.2.1	Coexistence scenarios	256
10.2.2	Preparing the AS/400 system for OneWorld	257
10.2.3	Overview of a coexistence installation plan	259
10.3	Converting WorldSoftware environment to coexistence	262
10.3.1	Installing OneWorld in a coexistence environment	266
10.3.2	Verifying the installation	267
10.4	Converting OneWorld environment to coexistence	269
10.4.1	Reading and converting data	270
10.4.2	Converting to a coexistence environment and saving the data	271
Chapter 11. Client Access Express and OneWorld		283
11.1	Client Access Express functions	283
11.2	Operations Navigator and Management Central	286

11.2.1	Operations Navigator functions	286
11.2.2	Operations Navigator tips and techniques	289
11.3	Client Access Express and Client Access/400	290
11.3.1	Benefits of Client Access Express	290
11.3.2	Transition from Client Access/400 to Client Access Express	292
11.4	ODBC	293
11.4.1	ODBC interface	293
11.4.2	ODBC components	293
11.4.3	Job flow of an ODBC connection with Client Access Express	295
11.4.4	ODBC configuration	296
11.5	License information	296
Chapter 12.	Operations	299
12.1	The OneWorld Installation Menu A98OWMNU	299
12.2	Setup tasks	300
12.2.1	Activating Service Director	300
12.2.2	Creating a problem escalation contact list	300
12.2.3	Building a system information binder	301
12.2.4	Determining key performance indicators	301
12.2.5	Creating an operations schedule	301
12.3	Setting up multiple job queues for use by OneWorld	301
12.4	Creating a command to set prestart job counts to greater than 1,000	305
12.5	Periodic tasks	305
12.5.1	Replicating TAM specifications on the AS/400 system	305
12.5.2	Starting and stopping the OneWorld system	307
12.5.3	Daily health check tasks	309
12.5.4	System management with Operations Navigator and Management Central	310
12.6	Cleaning up the OneWorld environment	320
12.7	Administrating the batch processes on the AS/400 system	321
12.7.1	Monitoring batch processes	321
12.8	Reviewing batch output files	323
12.9	Encoding the passwords of users who submit batch jobs	324
12.10	Integrated Netfinity Server operations	324
12.10.1	Checking the status of the Integrated Netfinity Server	324
12.10.2	Starting and stopping the Windows NT server	325
Chapter 13.	OneWorld package management	327
13.1	Package management planning	327
13.1.1	Understanding packages	327
13.2	Recommendations and considerations	329
13.2.1	Working with path codes	330
13.2.2	Object promotion and version control	331
13.3	Package deployment methodology	332
13.3.1	Workstation Installation	333
13.3.2	Package Deployment Instructions	333
13.3.3	Multi-Tier Deployment	334
13.3.4	Recommendations for sites using partial packages and JITI	334
13.3.5	Recommendations for sites using full packages and JITI	334
13.3.6	Related topics not covered in this chapter	335
13.4	Custom modifications	335
13.5	OneWorld modification rules	337
13.5.1	General rules for modification	338

13.5.2	Checking the environment before building packages	339
13.6	Defining and building a package	341
13.6.1	Working with package names	342
13.6.2	Overview of creating and deploying a package	342
13.6.3	Understanding the Package Assembly Director	343
13.6.4	Accepting default values	343
13.6.5	Assembling a package	344
13.6.6	Adding objects to a package	351
13.6.7	Activating a package	352
13.6.8	Defining a package build	353
13.6.9	Building business functions during a package build	354
13.6.10	Defining a package build	355
13.6.11	Tables used in a package build	365
13.6.12	Verifying package build errors	365
13.6.13	Strategy for full packages	369
13.6.14	Strategy for update packages	369
13.6.15	Understanding package deployment	369
13.6.16	Working with the Package Deployment Director	371
Chapter 14.	Printing	379
14.1	AS/400 printing	379
14.1.1	AS/400 printer attachments	379
14.1.2	AS/400 controllers and devices	381
14.1.3	AS/400 output queue	382
14.1.4	Printer files	382
14.1.5	Spooled files	382
14.1.6	Printer writer	385
14.1.7	Remote system printing	385
14.1.8	AS/400 NetServer	387
14.1.9	Client Access Express printing	387
14.1.10	Spool management	388
14.2	OneWorld printing	397
14.2.1	Creating a OneWorld printer	398
14.2.2	Testing printing in OneWorld	403
14.3	OneWorld printing rules, tips, and techniques	407
14.4	Report printing	408
14.4.1	Printing in batch	411
14.4.2	Print immediate functionality	412
14.4.3	Print filters paired to emulation for print stream	413
Chapter 15.	Security	415
15.1	AS/400 security architecture	415
15.1.1	Security levels	415
15.1.2	AS/400 security system values	415
15.1.3	AS/400 user profiles and group profiles	416
15.1.4	AS/400 object protection	417
15.2	OneWorld security	420
15.2.1	OneWorld user profiles and group profiles	421
15.2.2	OneWorld Security Server	423
15.2.3	OneWorld Security Workbench	429
15.2.4	Deployment Server directory and file security	433
15.2.5	OneWorld security recommendations	437
15.2.6	Adding and removing additional administrators	452

Chapter 16. Backup and recovery	455
16.1 AS/400 backup and recovery overview	455
16.1.1 OS/400 save and restore commands	459
16.1.2 Save-while-active	460
16.1.3 Journaling	461
16.1.4 ADSTAR Distributed Storage Manager	461
16.1.5 Backup Recovery and Media Services (BRMS)	461
16.2 Backup and recovery requirements	462
16.3 Saving in a J.D. Edwards OneWorld environment	462
16.3.1 OneWorld initial and periodic system saves	463
16.3.2 Weekly OneWorld saves	465
16.3.3 Daily OneWorld saves	467
16.4 Restoring in a OneWorld environment	468
16.4.1 Restoring the entire system	468
16.4.2 Restoring the OneWorld environment	469
16.4.3 System Managed Access Path Protection	469
Chapter 17. National language support	471
17.1 AS/400 national language support overview	471
17.1.1 Primary and secondary national language version	471
17.1.2 Language sensitive settings	472
17.1.3 NLV setting for cultural values	474
17.1.4 Multilingual and multi-system database support	475
17.1.5 System CCSID support	478
17.2 OneWorld national language considerations	479
17.2.1 Overview	480
17.2.2 Language installation	484
17.2.3 Installing an alternate language during OneWorld installation	484
17.2.4 Creating a language-only installation plan	494
17.2.5 Post-installation language verification	504
17.3 Tips and hints	505
17.3.1 Tips for implementing OneWorld SBCS languages	505
17.3.2 Tips for implementing OneWorld DBCS languages	507
17.4 OneWorld localization	510

Part 3. Faster, smoother, easier... 511

Chapter 18. The Integrated Netfinity Server and OneWorld	513
18.1 Introduction to the Integrated Netfinity Server	513
18.1.1 OS/400 V4R5 enhancements	515
18.1.2 OS/400 V4R4 enhancements	516
18.2 How the Integrated Netfinity Server works	516
18.2.1 Using system resources	516
18.3 Using OneWorld with the Integrated Netfinity Server	517
18.3.1 Benefits of Integrated Netfinity Server for OneWorld users	518
18.3.2 Integrated Netfinity Server considerations	519
18.3.3 Integrated Netfinity Server as a Deployment Server	519
18.3.4 Integrated Netfinity Server as Windows Terminal Server (WTS)	521
18.4 Installation	524
18.5 Startup procedures	524
18.5.1 Starting the Windows NT Server	524
18.5.2 Shutting down Windows NT server from the AS/400 side	525
18.5.3 Shutting down Windows NT server from the Windows NT side	526

Chapter 19. Data replication and data migration	529
19.1 Data replication	529
19.1.1 Planning a replication strategy	529
19.1.2 Data replication guidelines	530
19.1.3 IBM and third-party data replication solutions	530
19.2 Data migration	534
19.2.1 Data integrity	535
19.2.2 Data migration methodologies	536
Chapter 20. Server-to-server communication	545
20.1 Ways of connecting AS/400 OneWorld servers	545
20.2 OptiConnect/400 and virtual OptiConnect	545
20.2.1 Benefits	545
20.2.2 OptiConnect/400	546
20.2.3 Virtual OptiConnect	550
20.3 Network	551
20.4 Setting up an AS/400-to-AS/400 connection	551
20.4.1 OptiConnect considerations	551
20.4.2 Considerations in a non-OptiConnect environment	553
20.5 Connecting an AS/400 server with a non-AS/400 server	555
20.5.1 Two or more host servers running OneWorld code	556
20.5.2 Establishing a OneWorld service port	556
20.5.3 Defining host table entries	556
20.5.4 Modifying the Server Data Source Master table	557
20.5.5 Modifying the AS/400 server jde.ini file	559
20.5.6 JDBNet dependencies	560
Chapter 21. Performance management	561
21.1 Performance concepts	561
21.1.1 Queuing concepts	561
21.1.2 Response time curve	563
21.1.3 Client/server response time	564
21.2 Tuning	565
21.2.1 Client tuning	565
21.2.2 Network tuning	566
21.2.3 AS/400 server tuning	567
21.3 Performance analysis	597
21.3.1 Fixing performance problems: First steps	597
21.3.2 Performance analysis tools	605
21.3.3 AS/400 server performance measurement: Regular data collection	605
21.3.4 Server performance analysis using Performance Tools/400	608
21.3.5 AS/400 database query optimization tools	616
21.3.6 Database performance analysis: Operations Navigator	625
21.3.7 Database performance analysis: Centerfield Technologies Tools	648
21.3.8 Other AS/400 performance analysis tools	651
21.3.9 Client Access Express and client application PC tools	654
21.3.10 Network and communication analysis tools	660
21.4 AS/400 performance data collection procedures	663
21.4.1 Packaging performance evaluation data	670
21.5 OneWorld capacity planning with BEST/1	670
21.5.1 Introduction	670
21.5.2 AS/400 server modeling	672
21.5.3 Assumptions	672

21.5.4	Understanding the application environment	673
21.5.5	Selecting a capacity planning unit of measure	674
21.5.6	Determining growth objectives	675
21.5.7	Setting expectations	675
21.5.8	Measuring workload and system resource utilization	675
21.5.9	Collecting AS/400 performance data	676
21.5.10	Collecting OneWorld application workload data	677
21.5.11	Creating a model using measured data	679
21.5.12	Creating a BEST/1 model	680
21.5.13	BEST/1 workload components	690
21.5.14	Transaction	692
21.5.15	Working with BEST/1 models and results	693
21.5.16	Saving the BEST/1 model	699
21.5.17	Planning for growth	701
21.5.18	High priority threshold	705
21.5.19	Other considerations	706
21.5.20	Growth	707
21.5.21	Summary	708
Chapter 22. Problem identification and solution		711
22.1	Problem determination checklists	711
22.1.1	General problem determination checklist	711
22.1.2	Problems that affect system and application performance	712
22.1.3	Problems with upgrades	714
22.1.4	Installation problems	715
22.1.5	Problems related to OneWorld and WorldSoftware coexistence	716
22.2	Problem prevention	717
22.2.1	Informational APARs	717
22.2.2	Preventive Service Planning (PSP)	718
22.3	Problem identification aids	719
22.3.1	Job identification	719
22.3.2	Logs	730
22.3.3	Messages	738
22.3.4	Traces	746
22.3.5	OneWorld traces	754
22.4	Installation verification using PORTTEST	755
22.4.1	PORTTEST problem resolution	758
22.5	Tips and techniques	760
22.5.1	General troubleshooting hints	760
22.5.2	OneWorld UBEs work inconsistently	761
22.5.3	Debugging a Business Function (BSFN) on a logic server	762
22.5.4	CL program to start debug in QZDASOINIT jobs	766
22.5.5	Running OneWorld	766
22.5.6	SQL packages full, corrupted, or not being used	767
22.5.7	SQL packages beginning with an underscore or special characters	767
22.5.8	Client Access Express diagnostics	767
22.5.9	ODBC dropped connections	771
22.6	Remote support preparation	771
22.6.1	Preparing for IBM Remote Support	771
22.6.2	Preparing for J.D. Edwards remote support	772

Part 4. Appendixes	773
Appendix A. Adding a user auxiliary storage pool	775
A.1 Checking the disk configuration	775
A.2 Configuring a new user ASP	777
A.2.1 Starting Dedicated Service Tools	777
Appendix B. TCP/IP basic installation and configuration	781
B.1 Configuring TCP/IP on the AS/400 system	781
B.1.1 TCP/IP configuration using a command line interface	781
B.1.2 TCP/IP configuration using Operations Navigator	792
B.2 Connecting through a gateway or different network	814
B.2.1 Scenario	815
B.2.2 Route configuration	815
Appendix C. National language versions on AS/400 systems	817
Appendix D. Interprocess communication (IPC)	821
Appendix E. Support structure	823
E.1 Customer and business partner support	823
E.1.1 IBM support organizations and contacts	823
E.1.2 IBM AS/400 Support services	831
E.1.3 COMMON	834
E.1.4 J.D. Edwards customer support	836
E.1.5 Lotus and Tivoli contacts	838
E.1.6 QUEST (J.D. Edwards users group)	838
E.2 Business partner support	839
E.2.1 IBM Technical Sales Support	839
E.2.2 IBM Solution Assurance	839
E.2.3 PartnerInfo	840
E.2.4 IBM Product Support Services	840
Appendix F. Complementary products	843
Appendix G. Special notices	849
Appendix H. Related publications	853
H.1 IBM Redbooks	853
H.2 IBM Redbooks collections	854
H.3 Other resources	854
H.4 Referenced Web sites	856
How to Get ITSO Redbooks	861
IBM Redbook Fax Order Form	862
Glossary	863
Index	875
IBM Redbooks review	887

Preface

The J.D. Edwards OneWorld Implementation for AS/400 redbook offers a collection of knowledge gained from J.D. Edwards consultants and AS/400 professionals who have experience with WorldSoftware, OneWorld, and the AS/400 system. This is a detailed guide that explains the concepts, as well as specific tasks associated with implementation of OneWorld on the AS/400 system with *OS/400 V4R4*. This redbook is designed to assist technical people among J.D. Edwards OneWorld customers, OneWorld consultants, business partners, and IBM service representatives. It targets these professionals who are directly involved with the implementation of a total business solution consisting of AS/400 hardware, OS/400, DB2 UDB for AS/400 database, OneWorld solutions, and supplemental solution products.

The contents of this redbook are divided into four parts:

- Part 1 contains concepts and other basic knowledge, necessary to understand the structure, features, and functions of OneWorld and AS/400 system. This part is especially helpful if you are a beginner in the AS/400-OneWorld area or if you want to gain more insight into how OneWorld works on the AS/400 system.
- Part 2 describes the implementation tasks and techniques necessary to install and properly set up OneWorld on the AS/400 system. With the exception of Chapter 17, “National Language Support”, all OneWorld-AS/400 customers will face topics described in this part.
- Part 3 covers some advanced topics that will be interesting for readers who want to optimize their OneWorld AS/400 system to run smoother, faster, easier, and so on.
- Part 4 contains appendixes with complementary information.

The team that wrote this redbook

This is the second edition of the redbook produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center and J.D. Edwards in Denver. The authors of this second edition are:



Aco Vidovic is a Senior I/T Specialist at International Technical Support Organization Rochester Center on assignment from IBM Slovenia. Before joining the ITSO in 1999, he worked in the AS/400 area for 10 years in a variety of roles from second level technical support to marketing. In the ITSO, Aco writes extensively about ERP solutions implementation on the AS/400 system.



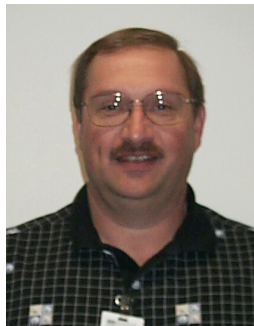
Dave Anderson is an AS/400 specialist currently working for J.D. Edwards in the Advanced Technologies Group. This group evaluates new technologies, performs benchmarks, and is deeply involved with CNC architecture deployment and performance tuning for the OneWorld product. Prior to working at J.D. Edwards, he spent 30 years working at IBM as a Systems Engineer specializing in the S/X, S/38, and AS/400. In this capacity, he also taught AS/400 performance for the IBM Rochester Lab.



Leann Anderson is a OneWorld Technical Support Consultant in World Wide Technical Support at J.D. Edwards. She has over 15 years of IT experience in a wide variety of capacities. Leann has worked at J.D. Edwards for two years specializing in database performance and tuning, while helping customers with tools design and system administration.



Jon Auger is the IBM Partner Lead for World Wide Advanced Technologies. He maintains the J.D. Edwards technical relationship with the IBM International Competency Center. His current responsibilities include technical project lead, problem escalation, and custom benchmarking. During his three years with J.D. Edwards, Jon has also performed WorldSoftware implementation, multiple language installations, and support regarding double-byte issues.



John Dulek has worked for IBM for 27 years and held various positions while working in Rochester, Minnesota. He has held such positions as test equipment operator, MVS/VM system operator, Level 2 S/38 and AS/400 computer programmer, and AS/400 course instructor. He currently works as the J.D. Edwards ERP Support Focal Point in the Rochester AS/400 Support Center.



Gerrie Fisk is a Senior I/T Specialist and has been with IBM for 24 years. She is a member of the IBM/J.D. Edwards International Competency Center, specializing in the AS/400 system. She is responsible for OneWorld benchmarks and sizing tests on the AS/400 system. Her previous assignments have been in the midrange arena as a Systems Engineer installing, supporting, and upgrading over 500 AS/400s and S/3Xs. She has also taught many AS/400 classes and seminars.



Rob Jump is a Senior I/T Specialist in the IBM/J.D. Edwards International Competency Center. His current work involves benchmarking and sizing tests for J.D. Edwards OneWorld on the AS/400 system. Rob has over ten years experience with System 36, 38, and the AS/400 system, working with installation, support, and communications. He is also a Certified Windows NT product specialist and has worked as a Windows NT and network consultant.



Bob Reynolds is an IBM certified AS/400 specialist currently assigned to the IBM/J.D. Edwards International Competency Center in Denver, Colorado. He has over 16 years of AS/400 and System 38 experience across many areas of Information Technology that include ERP implementation, application development, sales, systems engineering, business recovery, capacity planning, and project management.



Gottfried Schimunek is an IBM certified Consulting I/T Architect at the International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of application development and performance. Before joining the ITSO in 1997, Gottfried Schimunek worked in the Technical Support Center in Germany as a Consultant for IBM Business Partners and Customers, as well as for the IBM sales force.



Whit Smith has almost twenty years of experience working with IBM mid-range systems. He spent most of that time on the AS/400 system and its predecessor the S/38. Whit is currently focused on providing advanced technical support on the AS/400 platform for J.D. Edwards OneWorld software in the IBM Americas ERP Advanced Technical Support group.



Todd Thomsen is an AS/400 System, OneWorld Technical Support Consultant in World Wide Technical Support at J.D. Edwards. He has worked in a variety of IT related positions for the past six years. He has been working at J.D. Edwards with AS/400 System, OneWorld support for the last two years providing support in installing, upgrading, and administering OneWorld software as well as AS/400 system administration, tuning, and security administration.



Randall West is a Technical Consultant for the Advanced Technologies Group at J.D. Edwards. He has worked on the S/38 and AS/400 platforms for 20 years. During his 10 years at J.D. Edwards, he has served as an RPG and C developer, AS/400 system management specialist, and OneWorld implementation specialist. His current duties involve benchmarking various OneWorld configurations on the AS/400 system.



Ken Wettlaufer is a Senior AS/400 I/T Specialist and ERP Consultant for the ERP Practice of IBM Global Services in Canada. He specializes in the implementation of J.D. Edwards OneWorld on the AS/400, including coexistence with J.D. Edwards WorldSoftware. Ken has more than 23 years of experience with IBM in the field organization working with System/3, S/32, S/34, S/36, S/38, and the AS/400 system.

Thanks to the following people for their invaluable contributions to this redbook:

Jeff Amerine
Cliff Ard
Carolyn Barr
Debby Burns
Chantelle Cory
Pamela Coughlin
Jeff Crossman
Rick Dean
Mark Goedde
Michael Guerra
Tom Harger
Masood Khan
Kyle Kinder
Janet Knight
Randy Peterson
John Phillips
Mark Rask
David Rest

Luis Santillo
Dale Schmidt
Clayton Seeley
Terry Siebenneicher
Scott Sonczalla
Dave Tinucci
Diane Webster
Tatiana Wilson
J.D. Edwards Denver

Cyrill Belin
J.D. Edwards UK

Chung Kheng Wee
Desy
J.D. Edwards Singapore

Woo Wen Siang
Wong Wing Keng
Loh Yik Wai
BASC Malaysia

Mark Holm
Ken Peters
Centerfield Technology

Mathew Griffin
Pristine Concepts

Boyd Fenton
Patrick Moore
IBM/J.D. Edwards International Competency Center, Denver

Anne-Marie Stringer
IBM Canada

Rog Nolte
Neil Willis
Clark Scholten
Bob Schuster
IBM Rochester

Jarek Miszczyk
ITSO Rochester Center

Special thanks to the authors of the first edition of this redbook, who broke the ice and produced much of the material that can be still found in the second edition:

Jaejin Ahn
Linda Bingham Young
Jean Brown
Anil Madan
Stan Staszak
IBM

Dave Anderson
Michael Guerra
Mark Rask
Randall West
J.D. Edwards

Lee Walters
MAS

Devon Petty
CIBER

Thanks to following people for helping at the first edition of this redbook:

Heide Brace
Roger Chan
Jeff Crossman
David Damrell
Mark Goedde
Kathleen Heffernan
Sridhar Komakula
Betsy Pridmore
Buffy Ransom
David Rest
Tom Sidebottom
Terry Siebenneicher
Scott Sonczalla
Dan Stern
Yue Wu
J.D. Edwards & Company

James Denton
Selwyn Dickey
Gerrie Fisk
Boyd Fenton
Rob Jump
Chuck Masur
Patrick Moore
Rog Nolte
Neil Willis
IBM

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 887 to the fax number shown on the form.
- Use the online evaluation form found at <http://www.redbooks.ibm.com/>
- Send your comments in an Internet note to redbook@us.ibm.com

Part 1. Understanding the products

This part contains concepts and other basic knowledge necessary to understand the structure, features and functions of OneWorld and the AS/400 system. We recommend that you read this part if you are a beginner in the AS/400-OneWorld area or if you want to gain more insight into how OneWorld on AS/400 system works. This part includes the following chapters:

- Chapter 1, “Introduction to J.D. Edwards” on page 3, briefly introduces the J.D. Edwards company and their products.
- Chapter 2, “Introduction to IBM AS/400” on page 7, describes the AS/400 system key concepts, architecture, technology, and functions.
- Chapter 3, “OneWorld internal architecture” on page 31, describes OneWorld internal architecture, its building blocks and basics on how OneWorld works.
- Chapter 4, “Configurable Network Computing (CNC)” on page 47, explains what J.D. Edwards Configurable Network Computing is and how it is structured.
- Chapter 5, “OneWorld implementation designs” on page 71, explains the OneWorld implementation design concepts and provides examples of customer scenarios.
- Chapter 6, “Database” on page 83, reviews the concepts and terminology of DB2 UDB for AS/400, SQL, and query optimization tools and processes. This chapter also covers the OneWorld implementation of DB2 UDB for AS/400, including naming convention and methodology issues.
- Chapter 7, “Work management” on page 109, explains AS/400 work management, AS/400 Client/Server work management, and OneWorld work management.

Chapter 1. Introduction to J.D. Edwards

This chapter introduces you to J.D. Edwards and presents an overview of its WorldSoftware line and OneWorld solution. It also shows you the value and benefits that OneWorld can bring to a customer.

1.1 Company

In 1977 in Denver, Colorado, three men left the accounting world to form a software company that would specialize in midrange computing solutions. The three founders, Jack Thompson, Dan Gregory, and Ed McVaney, each lent a small portion of their own name for the company moniker.

In the early years, J.D. Edwards designed software for several small and medium-sized computers, eventually focusing on the IBM System/38 in the early 1980s. It was in this effort that J.D. Edwards pioneered the Computer Aided Software Engineering (CASE) development and design tool, which lends consistency across the broad range of J.D. Edwards' integrated applications.

As J.D. Edwards began to outgrow its headquarters space in Denver, it started opening branch offices in other parts of the country and in 1988, the company began to concentrate its efforts on international expansion. With the business continually growing, it became obvious that servicing a large number of customers was creating challenges. The company could either remain small and serve customers on an individual basis or, with a breakthrough in technology, it could become an industry leader in enterprise software. When Ed McVaney and Jack Thompson began to design and implement the WorldSoftware application package, they provided the pathway to success. By the late-1980s, J.D. Edwards was recognized as an industry-leading supplier of applications software for the highly successful IBM AS/400 computer, a direct descendant of the System/38.

Today, J.D. Edwards is a publicly traded company that has more than 5,500 customers in over 100 countries and more than 5,600 employees in 48 offices worldwide.

The company posted 1999 fiscal revenues of \$944 million and has grown at a rate of 40% per year for over 10 years. J.D. Edwards attributes much of its success to a corporate culture that emphasizes quality at all levels.

Additional information about J.D. Edwards can be obtained by contacting the company by telephone at 1-800-727-5333 or on the Internet at:

<http://www.jdedwards.com>

1.2 Products

J.D. Edwards is a provider of Idea to Action enterprise applications. These applications encompass flexible, integrated software for distribution, finance, human resources, manufacturing, and supply chain management. The company's software operates in multiple computing environments, it is Java and HTML enabled. J.D. Edwards enables Idea to Action with ActivEra, a customer-centric product and technology solution that allows companies to change enterprise software after implementation as their business needs demand. ActivEra extends

the capabilities of the company's existing OneWorld and J.D. Edwards SCOREx products.

The list of J.D. Edwards applications includes:

- Foundation Suite
- Financial Suite
- Logistics/Distribution Suite
- Services Suite
- Manufacturing Suite
- Architecture, Engineering, Construction, Mining, and Real Estate Suite
- Energy and Chemical Suite
- Payroll Suite
- Human Resources Suite
- Customer Service Management Suite
- Government, Education, and Not-for-Profit Solutions
- Utility and Energy Solutions

These applications are available as a part of two application packages: WorldSoftware and OneWorld. This redbook mostly covers OneWorld. Chapter 10, "WorldSoftware and OneWorld" on page 247, describes how WorldSoftware and OneWorld can coexist in the same environment and how customers can migrate from WorldSoftware to OneWorld.

1.2.1 WorldSoftware

WorldSoftware is a host-centric application running on the AS/400 system, driven by native AS/400 RPG code. This is a stable and flexible solution used by more than 5,500 customers. WorldSoftware's global popularity has endured in the ERP marketplace for over a decade. Inherently flexible and easy to use, WorldSoftware is readily adaptable to a customer's situation, allowing them to:

- Selectively mix, match, and integrate software applications from among its diverse industry product suites.
- Easily tailor it to ongoing business, local, and organization-specific requirements.
- Add WorldVision, a graphical user interface to traditional "green-screen" interface.
- Optionally run it alongside OneWorld, J.D. Edwards' network-centric solution, on the same system, and thus gradually incorporate other computing platforms into your network.

1.2.2 OneWorld

With the June 1996 introduction of OneWorld, J.D. Edwards achieved a technological breakthrough. Building on the Computer Aided Software Engineering (CASE) technology pioneered in the 1980s, OneWorld is a platform independent application with an integrated toolset, which allows customers to configure their systems and applications as their needs change. It continues the proven database architecture of the WorldSoftware products, and thus leverages the years of industry application experience J.D. Edwards has acquired.

OneWorld is a network-centric solution that separates business rules from the underlying technology. As new technologies evolve, customers can add them to the framework without disrupting their ongoing business.

1.2.2.1 Why OneWorld

OneWorld is architecture neutral, so it can run on many different platforms. Here is a list of benefits for OneWorld customers:

- It integrates a customer's supply chain by using the Internet and the organization's intranet to keep customers in sync with their employees, customers, and suppliers.
- Its interoperability lets customers leverage their existing investments in hardware, databases, and software by integrating their legacy and third-party products.
- It adapts easily to different languages, currencies, reporting provisions, and technology standards, simultaneously.
- With its Windows-like, graphical user interface (GUI), it is easy to use. Using a mouse, users can simply point-and-click and drag and drop.
- By the end of year 2000, J.D. Edwards plans to add euro functionality to OneWorld.

OneWorld's toolset uses business logic, not symbols or syntax, to drive the modification process. This allows changes to business specifications, while the toolset automatically regenerates the appropriate object code. Reprogramming does not require a traditional programmer. Modifying applications, balancing processing loads, running reports, and building graphical user interfaces can be accomplished without writing code. Adding hardware and databases does not require bringing customer's business to a halt. Since modifications are made with the same toolset used to build OneWorld, they are integrated into OneWorld. New releases are automatically incorporated, training time is saved, and the user interface remains consistent. OneWorld also allows customers to build highly flexible workflow solutions and execute both predefined and ad hoc processes in the organization.

J.D. Edwards is currently releasing a Web version of OneWorld, which allows companies to run applications over the Internet using HTML or Java. Their interoperability strategy includes wide-ranging APIs (both industry standard and OneWorld-bases), object standards, interoperability processware solutions, and other methods.

Using OneWorld, customers can build an extended solution while reducing much of the complexity necessary to do so. The OneWorld architecture supports interfaces to third-party application solutions, so customers can leverage the strengths and benefits of that architecture even as they incorporate existing and future applications and data.

1.2.2.2 OneWorld technology

OneWorld embraces change with its modular architectural foundation. J.D. Edwards segmented their information processing into five functional elements:

- Database
- Data warehouse
- Business objects
- Reporting
- Graphical user interface (GUI)

Customers can link these elements in a variety of configurations, from one tier, with every element running on a stand-alone PC, to many tiers. They can distribute the elements geographically, departmentally, or administratively. Customers can configure and reconfigure in the field, on the fly, as requirements change. They can add new servers, including Web servers, without having to rewrite applications for the new machine.

OneWorld allows customers to distribute their enterprise applications to employees, business partners, and customers using Web-based technology, without rewriting your applications. One software version supports both client/server and Internet modes.

For details on OneWorld technology, refer to Chapter 3, "OneWorld internal architecture" on page 31, Chapter 4, "Configurable Network Computing (CNC)" on page 47, and Chapter 5, "OneWorld implementation designs" on page 71.

Chapter 2. Introduction to IBM AS/400

This chapter provides an overview of the AS/400 platform: key concepts, architecture, technologies, and functions.

The IBM AS/400 system has a long and successful history worldwide. At the end of last millennium, there were more than 700,000 AS/400 systems installed in over 150 countries. There are more than 30,000 AS/400 business applications available worldwide, including over 2,700 applications for e-business. The reason for this success is founded in six basic factors, which are described in the following list:

- **Architecture:** The AS/400 system has a layered architecture that is divided into the actual user interface (operating system OS/400) and a Technology Independent Machine Interface (TIMI). This architectural concept has allowed the AS/400 system to undergo several fundamental technology changes while protecting the customer's investment in information technology. Refer to 2.1, "AS/400 architecture" on page 9, for more information.
- **High level of integration:** One of the key factors that differentiates the AS/400 system from other systems is the level of hardware and software integration. For example, in a UNIX environment, customers are required to select software components from different vendors (operating system, database, system management software, and so on). They are also required to integrate them to build a working environment for OneWorld. The AS/400 system takes a different approach. Hardware, microcode, the operating system, and IBM middleware are tightly interlaced allowing maximum exploitation of all available computing resources. The operating system includes a complete range of "licensed programs" (middleware) and offers the highest level of integration. By effectively reducing the extent of integration required to be performed during implementation, the AS/400 system approach minimizes implementation costs, increases reliability, and provides customers with the highest level of "ease-of-use".
- **Interoperability:** The AS/400 system offers a wide range of communication capabilities and functions that enable the AS/400 system to communicate with other IBM and non-IBM systems.

Communications protocols supported by the AS/400 system include:

- TCP/IP
- SNA
- OSI
- NetBIOS
- IPX/SPX
- AnyNet

Note

J.D. Edwards OneWorld uses the TCP/IP protocol.

Data link protocols supported on the AS/400 system include:

- Ethernet
- Token-Ring
- Asynchronous Transfer Mode (ATM)
- ISDN

- FDDI
- Synchronous Data Link Control (SDLC)
- X.25
- **Client/server capability:** The AS/400 server can operate with following client platforms:
 - Microsoft Windows 3.1, Windows 3.11, Windows 95, Windows 98, Windows NT, and Windows 2000
 - UNIX (IBM-AIX, HP-UX, SUN-SPARC)
 - Apple Macintosh
 - IBM OS/2
- **Scalability:** AS/400 systems cover a wide range of performance capacities. AS/400e servers 1XX, 2XX, 7XX, and 8XX provide many processor options to choose from based on the performance requirements. At the high end of these servers, the AS/400e server 840 with 24 central processors provides 330 times the performance boost over the smallest model 250. Figure 1 shows performance span of newest AS/400 models 250, 270, 820, 830, and 840, measured in Commercial Processor Workload (CPW).

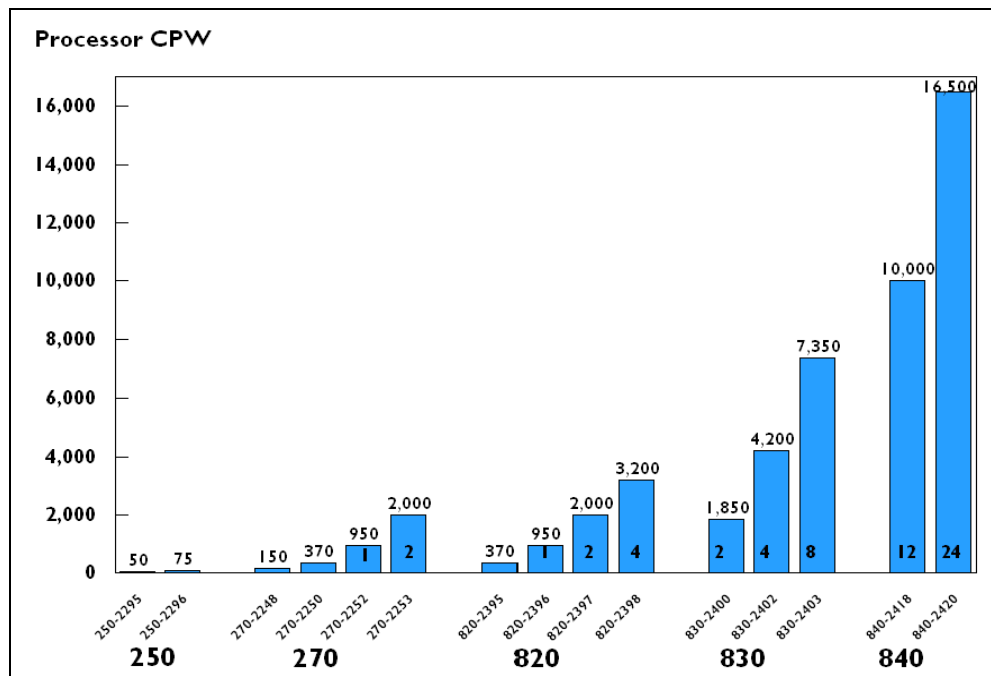


Figure 1. AS/400 scalability

For J.D. Edwards OneWorld customers, this means an AS/400 system can support environments in the size of few users to thousands of users. Many of these AS/400 system models are field-upgradable to more powerful models, which provide an exceptional degree of customer investment protection. It is one of the contributing factors to the AS/400 system's low cost of operation in the long term.

- **Price/performance:** Many independent analysts have confirmed that the AS/400 system represents a cost-effective platform in the long run. The AS/400 system's extensive integration yields significant cost advantages.

2.1 AS/400 architecture

One of the key differentiators and contributors to the commercial success of the AS/400 system is its integrated architecture. This integration includes the hardware, operating system, and middleware. It is illustrated in Figure 2.

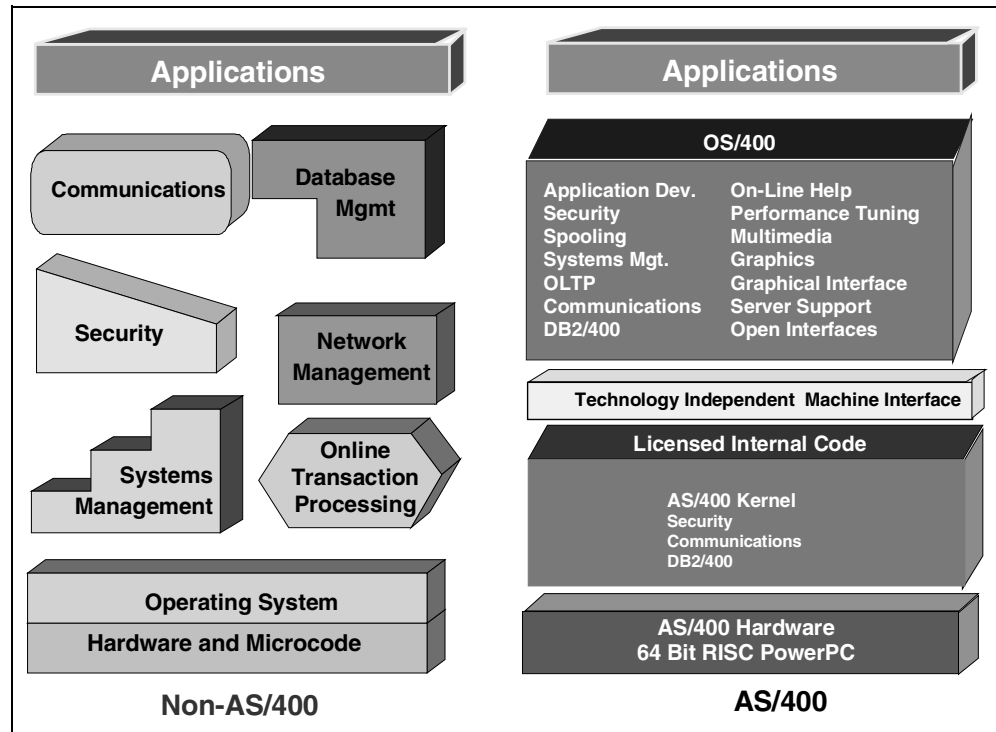


Figure 2. AS/400 System: An integrated system

The IBM engineers designed and built the AS/400 hardware to integrate it with the operating system (OS/400). OS/400 includes a wide array of middleware components including a relational database, an Internet server, system management facilities, built-in facilities for change management, backup and recovery, and print handling. It also includes communication capabilities (such as TCP/IP, SNA, and IPX), a POP3 mail server, and client support (Windows clients, MacIntosh, OS/2, and some UNIX clients). As a part of OS/400, these middleware products share the same packaging, user interface, installation, maintenance, security, and management facilities.

The integration and testing of these components in the IBM lab ensures that the AS/400 system does not need a skilled systems integrator at the customer site. The AS/400 system arrives ready to start. This integration is the reason why the AS/400 system delivers a low cost of ownership.

2.1.1 High level machine interface

Technology Independent Machine Interface (TIMI) shown in Figure 3 on page 10 is a design boundary that separates the hardware with Licensed Internal Code

(LIC) from the operating system (OS/400). This permits the instructions used by the compilers of high-level languages to be generic and machine (hardware) independent. These instructions are translated to a specific hardware instruction set as part of the back end of the compilation process. Hardware dependencies are absorbed by LIC (internal microcode).

The ability of the AS/400 architecture to accommodate the smooth transition from Complex Instruction Set Computing (CISC)-based processors to Reduced Instruction Set Computing (RISC)-based processors is one example of *technology independence* provided by TIMI.

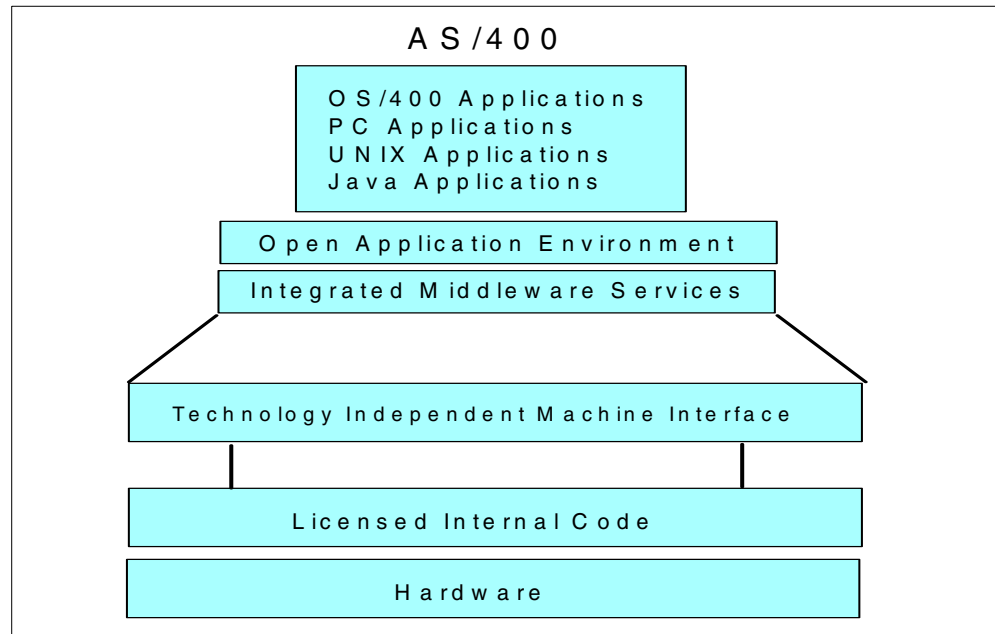


Figure 3. AS/400 advanced application architecture

2.1.2 Hierarchy of microprocessors

In addition to its central system processor (CPU), the AS/400 system has a range of other processors, each dedicated to a particular input/output (I/O) device type. A single large AS/400 configuration can have well over 200 processors.

When the CPU (which itself can be comprised of up to 24 separate processors) encounters a request for data to be read from or written to a I/O device, such as disk, it delegates that request to the particular microprocessor dedicated to that I/O device. While the I/O processor processes the I/O operation, CPU can continue executing another application program. Since CPU is much faster than I/O devices (nanosecond - 10^{-9} second is the unit of time used to measure CPU clock speed, as opposed to I/O operations measured in milliseconds - 10^{-3} second), this means that by delegating its work to I/O processors, CPU avoids unnecessary wait time.

This design provides the AS/400 system with its outstanding performance in the commercial, transaction-based, environment which is I/O-intensive, rather than compute-intensive.

This design also gives the AS/400 system an elegant method of integrating diverse environments into a single, harmonious customer solution. The microprocessors that look after a particular I/O device are accommodated on I/O cards that fit into slots on the AS/400 system bus. One of these cards can be the Integrated Netfinity Server. This PC on a card fits into an AS/400 slot and enables the AS/400 system to run PC software, such as, Windows NT server or Novell Netware. For more information on Integrated Netfinity Server, refer to Chapter 18, “The Integrated Netfinity Server and OneWorld” on page 513.

2.1.3 Object-based architecture

Objects are the means through which information is stored and retrieved on the AS/400 system. This concept is different from typical byte-string and file manipulation in many other systems. Object orientation is part of the architecture and affects both the operating system implementation and high-level language interaction with the system.

As previously mentioned, the TIMI is a boundary (set of instructions) that separates the hardware and LIC from the operating system. The AS/400 system instructions have an operation code and operands. Unlike many other systems, the operands are objects, and the instructions act on objects.

Objects have operational characteristics and a defined set of operations that can be performed on them. Objects are addressed through 16-byte pointers (eight bytes are used for a machine address. The other eight bytes are used for information about the object pointed to and for reserved space). In addition to providing addressability to the object, pointers provide access to the associated storage and are used to ensure data integrity and security.

Below the TIMI, LIC provides a tag bit for each *quadword* (16 bytes that must be aligned on a 16-byte boundary) within main storage. This bit is not addressable by the TIMI instructions used to address storage (that is, programs above the TIMI have no direct access to the tag bit). The bit identifies quadwords in storage containing TIMI pointers. The tag bit is turned on by LIC when a pointer is set and turned off by the hardware anytime the quadword is modified. This procedure allows the system to detect invalid pointers and prevent illegal use of a pointer. An attempt to subsequently use this data as a pointer results in an exception and the instruction is not completed. It is not possible to counterfeit a pointer or to modify a pointer in an invalid way. This provides for increased integrity of the machine against intentional or unintentional software actions.

2.1.4 Single level storage

One of the key resources on any computer system is the disk space that stores the data, applications, and programs. The management of this storage space is critical to having an efficient and high performance server installation. Storage management on the AS/400 system is automated. The AS/400 system takes care of selecting the physical disk drive (sometimes referred to as Direct Access Storage Device (DASD)) to store the data, spreads the data evenly across the disk arms and continues to add records to files until specified threshold levels are reached.

AS/400 system storage management uses single level storage. With single level storage, there is a single, large address space for all storage (both main storage and disk storage). Storage is addressed using a 64-bit (8-byte) addressing

structure. When an object is created, it is defined in a unique virtual address space. There is a single page table (sometimes referred to as a *page directory*) that maps all virtual addresses to corresponding physical addresses.

Application programs on an AS/400 system are unaware of underlying hardware characteristics because of the TIMI. In the same way, they are also unaware of the characteristics of any storage devices on the AS/400 system because of single level storage.

As with TIMI, the concept of single level storage means that the knowledge of the underlying characteristics of hardware devices (in this case, the hardware storage devices—main storage and disk storage) reside in the System Licensed Internal Code (SLIC). Programs work with objects, and objects are accessed by name, never by address. No user intervention is ever needed to take full advantage of any storage technologies.

The AS/400 system can address the number of bytes that 64 bits allows it to address. For example, 2^{64} is 18,446,744,073,709,551,616. Therefore, the AS/400 system can address 18,446,744,073,709,551,616 bytes, or 18.4 quintillion bytes.

Single level storage enables another extremely important AS/400 benefit, *object persistence*. Object persistence means that the object continues to exist in the memory system forever. An ordinary machine requires that information be stored in a separate file system if the information is to be shared or if it is to be retained for a long time. The persistence of objects is extremely important for future support of object-oriented databases. Objects need to continue to exist even after their creator goes away. The AS/400 system is uniquely positioned to exploit this characteristic of object persistence, where ordinary systems use a less-elegant mechanism that requires them to store their persistent objects in a separate file system with all the attendant performance implications.

The benefits of single-level storage include:

- All applications use one address space on the AS/400 system. Switching processes (for example, from database to Internet) requires little time. There is no need to flush out one address space and bring in another one.
- With single-level storage, there is no need for swap or paging space. When data needs to be moved out of memory, it goes directly to its permanent location on disk. It is not written first to a swap file and then its permanent location. AS/400 system disk space is not wasted for swap files.
- With single-level storage, all information is in one large address space regardless of it being in memory or on disk. All objects appear to be in memory. When a program wants to reference data, it simply references it by its address. There is no need for the program to translate addresses or go through a lengthy process to determine where the file exists on disk.

With disk and memory managed through a common address space, the system can take advantage of increased memory without needing any complex memory management activity.

- Single-level storage also holds tremendous promise for object-oriented programs such as Java. With the ability to have shared persistent objects, the AS/400 system can eliminate the processing required on other systems to hydrate and dehydrate objects.

Single-level storage is a key reason why the AS/400 system is easy to develop applications for, can support multiple applications, and is scaled to support a large number of users. Single-level storage, combined with the automated storage management capabilities, make the AS/400 system an easier system to manage.

The AS/400 system customer can manage disk space on the AS/400 system as a single entity. The disk space can be used by all of the applications. The system determines when and where to extend individual files and spreads the data across the disk arms to maximize performance. The system keeps track of the disk usage and warns the AS/400 system administrator when the disk space in the system auxiliary storage pool (ASP) reaches a specified threshold value. Advanced AS/400 system installations can deploy other disk management options to increase control of the space management process.

OS/400 has sophisticated space allocation routines to create and store information in the right size spaces. These routines look for the right sized block as opposed to selecting the first available block, regardless of size. The objective of these routines is to effectively use the disk space by reducing file and free space fragmentation. The AS/400 system automatically consolidates disk space fragmentation where possible and also has a separate disk fragmentation utility.

2.2 64-bit technology

Today, AS/400 systems use 64-bit PowerPC processors that have 64-bit registers. OS/400 is a 64-bit operating system, and its instructions perform 64-bit operations. Functional units can sort, compare, load, add, and subtract values that are 64-bit. Storage addressing uses 64-bits. Data paths move data from one location to another, 64-bits at a time (for example, from the cache to the processor).

OS/400 middleware is also 64-bit, including the database (DB2 UDB for AS/400). This middleware exploits the 64-bit support in the operating system and hardware.

All AS/400 system applications exploit the new 64-bit architecture. Older AS/400 system applications are automatically migrated to run under 64-bit RISC technology. They do not have to be recompiled or rewritten to support 64-bit addressing, 64-bit data paths, or 64-bit registers.

Complete 64-bit support is a key enabler for delivering performance to demanding applications, such as data warehousing, data mining, or Enterprise Resource Planning (ERP) software like J.D. Edwards OneWorld.

2.2.1 Copper chip technology

The newest AS/400 models 270, 820, 830 and 840 use Pulsar and Istar processors with copper chip technology. This is the sixth generation of AS/400 PowerPC 64-Bit RISC processors. Northstar processors used in prior AS/400 systems deploy aluminum for on-chip wiring.

Copper's better conductivity permits thinner wires to be used, which enables the transistors to be packed closer together. This new denser technology allows for

additional micro architecture methods to improve performance. Denser processor technology also permits more on-chip cache.

2.2.2 Silicon on Insulator (SOI)

Istar processors used in AS/400 Model 830s, 840s, and some of the Model 820s are the first processors in the industry that use the Silicon on Insulator (SOI) processor technology. The addition of SOI alone can increase performance up to 20 to 30 percent beyond the use of copper by protecting the millions of transistors on a chip with a blanket of insulation, reducing harmful electrical leakage that wastes power.

The transistors are built within and on top of a thin layer of silicon that is on top of an insulating layer. The insulating layer is fabricated by implanting a thin layer of oxide beneath the primary silicon surface of the wafer. This allows the high-end AS/400 Model 840 to be 3.6 times faster than the previous high-end AS/400 Model 740.

2.3 Hot-plugging

The term “hot-plug PCI” is used by the computing industry for a wide range of purposes. Primarily, hot-plug PCI refers to the technology that allows users to install and remove Peripheral Component Interconnect (PCI) devices while the computer is running.

PCI hot-plugging is available on the newest AS/400e servers with V4R5 on all 8xx models and some of the 270 models. Hot-plugging in the AS/400e is made possible by power control to individual card slots, so that PCI input/output processors (IOPs) or input/output adapters (IOAs) can be added, removed, or replaced while the system remains active. In most cases, IOA configurations can be changed while other IOAs on the same IOP remain operational.

2.4 High speed link (HSL)

In the hardware of a server, there are many factors that affect performance. Bandwidth and configuration of the I/O system have a big effect, especially in commercial environment, as does the network connecting the computers. New AS/400 processors supported by OS/400 V4R5 with more on-chip cache, a faster memory bus, a super-fast data cross-bar switch complex (with speeds up to 36 GB per second), faster disks, and much faster I/O processors and I/O adapters necessitate a new data transportation mechanism to keep processors of the AS/400 busy.

HSL is capable of transporting massive amounts of data from the I/O towers with their attached devices to the processor at a speed of up to 1 GB per second, which is up to 10 times faster from the previous solution.

2.5 Integrated relational database (DB2 UDB for AS/400)

The integrated relational database manager has always been one of the most significant facilities that the AS/400 system has provided. Relying on a database manager integrated into the operating system means that almost all of the user data on the AS/400 system is stored in a relational database and that access to

the database is implemented by the operating system itself. Some database functions are implemented at a low level in the AS/400 system architecture and some are even performed by the hardware to improve performance.

The integrated database, DB2 Universal Database for AS/400 (DB2 UDB for AS/400), provides stability and compatibility of previous releases of the AS/400 database with the standards-based technology required for a heterogeneous computing environment. DB2 for OS/400 provides compliance in the area of standards compliance coupled with advanced function, distributed capabilities, and performance. DB2 for OS/400 provides the following features:

- Structured Query Language (SQL) standards conformance supplies the industry standard database access language conforming to the IBM SQL Version 1, ANSI X3.135.1992, ISO 9075-1992, and FIPS 127-2 standards. Support is provided for embedded static, dynamic, and extended dynamic SQL, together with IBM's Distributed Relational Database Architecture (DRDA), Microsoft's Open Database Connectivity (ODBC), and Apple's Data Access Language (DAL). A Call Level Interface (CLI) server mode is also provided that allows developers to write applications that do database serving for multiple users.
- Encoded Vector Indexes (EVI) can be created using SQL. EVIs cannot be used to order records, but in many cases, they can improve query performance.
- Declarative referential integrity preventing conflicting data from being entered in the database.
- Stored procedures allowing the distribution of application workloads between a client and an Application Server.
- Triggers that cause automatic program execution before and after database modifications.
- Two-phase commit transaction management to allow access to multiple heterogeneous databases simultaneously.
- Data replication automatically in distributed DB2 family environment.
- System-wide database catalog allowing applications to query information concerning all objects on a system using a single system catalog.
- Multiple-level concurrency control providing read stability, cursor stability, uncommitted read, and no commit isolation levels.
- National language support to store data in a preferred language, character set (single and double byte), and a sort sequence.

For more information, refer to Chapter 6, "Database" on page 83.

2.6 Data management

AS/400 system applications communicate with system resources by using objects. For example, they use *display file objects* to communicate with traditional, Non-Programmable Terminal (NPT) display devices, *printer file objects* for printer devices, and *database file objects* for AS/400 database files.

The AS/400 system display file has a record format that contains screen field definitions and other screen control parameters. The screen field definitions can

optionally obtain their attributes from a database file field through a reference function.

Data Description Specification (DDS) is used to define the display file record and database file record and field attributes. When a standard DDS file is used, AS/400 system workstation management is responsible for formatting the device dependent data streams on output and unformatting on input. The AS/400 system database file object defines record field definitions and optionally identifies key (index) fields. The database field definitions can also optionally be defined by referencing other database file field definitions.

In most AS/400 system applications, the workstation and database field definitions are integrated during the compile (create program) step. This is referred to as externally described file support.

In addition to display, printer, and database interfaces, the AS/400 system provides message queues, data areas, data queues, and other interfaces, including Portable Operating System Interface Standard (POSIX) compliant.

2.7 Security, user profiles, and authority management

The AS/400 system has an outstanding reputation as a mission-critical commercial server. Part of this reputation is due to the security of the AS/400 system. OS/400 has a single integrated security model. This security model is shared by the operating system, database, mail, systems management, the Internet, and communications functions. You can add a user once to OS/400, and they can have access to all the appropriate components. These include Windows NT, OS/2 Warp Server and Novell NetWare on the Integrated Netfinity Server or Domino for AS/400 running natively on the AS/400 system.

OS/400 has a single directory, which is the system distribution directory. This directory is used by OS/400 e-mail (POP3 and OfficeVision) and Domino for AS/400.

The AS/400 system has a C2 security certification from the U.S. Department of Defense and is the only computing system to have received certification for both a database and operating system as a unit. The AS/400 system configuration that was certified was a functional configuration that included non-programmable terminals.

The AS/400 system has an object-based system architecture (see 2.1.3, "Object-based architecture" on page 11). Everything is an object (including data, programs, files, and so on) and is subject to object rules. These rules describe what can be read, changed, deleted, or added, and by whom or what. The AS/400 system checks those rules before it touches any object. This basic and underlying capability provides AS/400 system customers with a powerful security capability within their system. In addition, program objects have special rules that do not allow them to run unless they are properly encapsulated by the system. This makes the AS/400 system extremely virus resistant. AS/400 system objects, including programs, files, and databases, are protected by the OS/400 security mechanisms. When correctly implemented, this protection cannot be compromised by someone who merely has access to the machine.

System authorization management is based on user profiles that are also objects. All objects created on the system are owned by a specific user. Each operation or access to an object must be verified by the system to ensure the user's authority. The owner or appropriately authorized user profiles may delegate to other user profiles various types of authorities to operate on an object. Authority checking is provided uniformly to all types of objects within a file system.

The object authorization mechanism provides various levels of control. A user's authority may be limited to exactly what is needed.

For more information on security and user profiles, refer to Chapter 15, "Security" on page 415.

2.8 Integrated file system (IFS)

To enable the AS/400 system as a *server-of-choice* serving the major clients such as PC clients, UNIX clients, and the AS/400 system clients themselves, it was necessary to implement each client's file system on the AS/400 system as natively as possible.

The integrated file system (IFS), is a part of OS/400 that supports input, output, and storage management similar to the PC and UNIX operating systems. At the same time, it provides an integrated structure over all information stored in the AS/400 system. The key features of the IFS are:

- Support for storing information in stream files
- A hierarchical directory structure
- An integrated interface to access stream files, database files, documents, and other objects stored in the AS/400 system

A diagram of the IFS is shown in Figure 4.

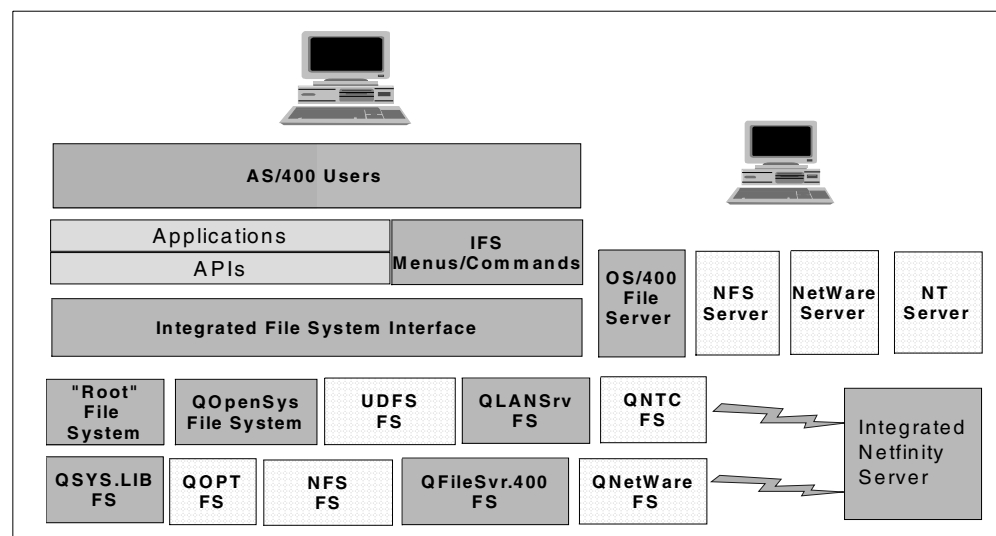


Figure 4. Integrated file systems

For more information on IFS, refer to *OS/400 Integrated File System Introduction Guide*, SC41-5711.

2.9 Logical Partitioning (LPAR)

Logical Partitioning (LPAR) allows multiple independent servers, each with its own processors, memory, and disks, to run within a single symmetric multiprocessing AS/400 system.

The end objective of LPAR, released with OS/400 V4R4, is to provide users with the ability to split a single AS/400 system into several independent systems capable of running applications in multiple, independent environments simultaneously. For example, Logical Partitioning makes it possible to run an application such as OneWorld, using different sets of data on separate partitions, as if it was running independently on separate physical AS/400 systems.

2.9.1 LPAR scenarios

The range of scenarios possible with Logical Partitioning is limited only by customer's imagination. The following sections explore a few of the most promising scenarios.

2.9.1.1 Multiple production environments

A company that offers outsourcing services to its customers can use a multiprocessor high-end AS/400 system with enough resources to create several completely independent logical partitions to meet customer requirements. These partitions can operate as independent machines. For example, some partitions may run client/server workloads, while others run purely interactive workloads, and still others may run mixed workloads. The permutations are endless.

2.9.1.2 Mixed production and test environments

Generally, production and test environments should be kept separate. LPAR achieves this objective in a simple way.

One or several partitions can be allocated for production only. Others are set up to handle application testing, either to validate application changes or to test applications under new releases of system software.

Without LPAR, the only practical way of testing a new version of an application would be to purchase additional hardware and software. However, the disadvantage may be that the hardware and software may not be required subsequently. Note that system software is currently charged at the physical machine level. Therefore, it can be run on several independent partitions, without an additional charge.

Partitioning provides a solution by making it possible to use an independent partition for testing. When testing is over, the resources allocated to this partition can be returned to the production partition or elsewhere as required.

2.9.1.3 Implementing an archiving solution

In this scenario, the customer has a very large database containing a mix of live and historical data. Most of the interactive and batch jobs use the live records on the database. However, the customer needs to access the historical data both interactively and in batch mode from time to time, with reasonable response times. Furthermore, they also need to reduce their backup time, without jeopardizing data security. Assuming that the application can distinguish between

live and historical data, one possible solution is to create a three-partitioned configuration as shown in Figure 5.

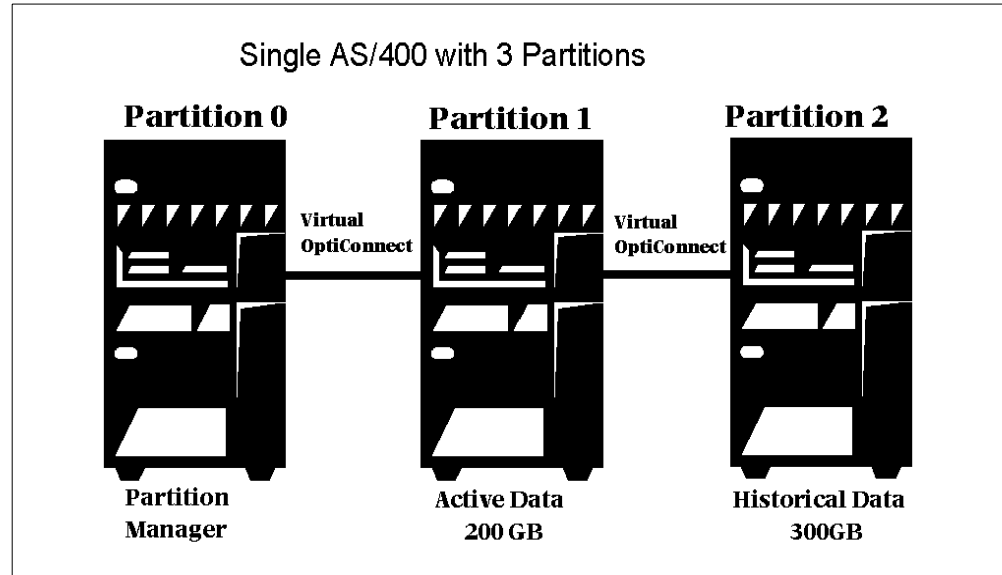


Figure 5. Setting up an archiving solution

The outcome of this implementation would be:

- Faster updates of live data because of reduced database index tables.
- Seamless access to historical data through high-speed inter-partition communication.
- Reducing the size of the database also reduces the size of the database index tables and, therefore, the amount of work the CPU has to do to update these tables.
- Overall improved response times.
- Reduced backup requirements. Only live data needs to be backed up at frequent intervals. Backing up archived data needs to be performed only at the time when live data transitions to archived status.
- Report consolidation through DB2 Multisystem or Remote SQL over high-speed inter-partition communication.

2.9.1.4 Server consolidation

Today's AS/400e server offers over 300 times the performance range from the smallest to the largest servers. Many companies are taking advantage of the high degree of scalability by consolidating existing AS/400e applications onto fewer systems, often in regional data centers.

Still many companies are attracted to the concept of distributed computing, in particular on the AS/400, UNIX, and Windows NT servers. The attraction of distributed servers is due to the low initial investment costs and the flexibility of deploying applications customized to the needs of specific departments. However, the realities of managing distributed UNIX and Windows NT platforms proved far more difficult and costly than initially imagined.

In response, many companies are now investigating opportunities to consolidate their decentralized and distributed systems to regain both financial and administrative control of their IT resources. This trend in the industry is referred to as *server consolidation*.

With its strengths as both an enterprise and distributed server, the AS/400 system has a unique position as a server consolidation platform for AS/400 consolidation, Domino for AS/400, and Windows NT Integrated Netfinity Server.

For more information on server consolidation, see the IBM Web site at:

<http://www.as400.ibm.com/sc>

2.9.1.5 Multi-database applications

Some businesses may require running the same application against independent databases. It may be that, because of naming conventions or other internal restrictions, the application cannot be run against multiple independent databases on the same physical machine. The solution before Logical Partitioning would be either to purchase additional hardware or modify the application.

With Logical Partitioning, each instance of the application can be run within its own independent partition. In addition, communications between the independent partitions can be used for consolidated reporting if required.

2.9.1.6 Minimizing backup and recovery windows

The imperatives driving today's businesses demand total machine availability, 7 days-a-week, 24 hours-a-day (7 x 24). On the other hand, data security demands that a foolproof backup strategy be in place to meet unforeseen contingencies. A third ingredient in this situation is the relentless growth of databases as businesses become more complex. Logical partitioning can provide one solution to balance these conflicting needs.

Let us look at a scenario where backup is becoming so time consuming that the production window is dwindling fast. Figure 6 illustrates a possible solution to minimize backup of the production database. Incidentally, this scenario also facilitates recovery.

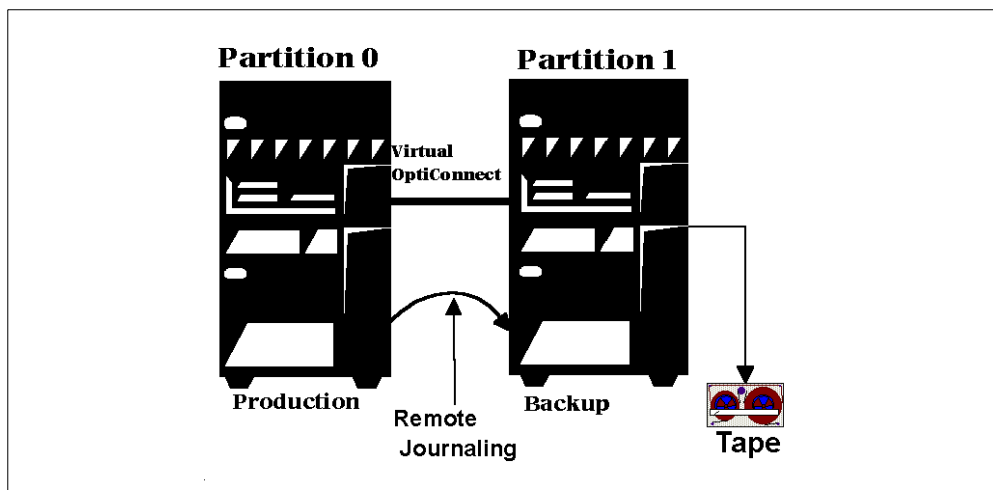


Figure 6. Minimizing backup

In this scenario, production is running on partition 0, and all updates are replicated on partition 1, using remote journaling. At preset intervals, the partition 1 update is suspended, and the partition database is backed up. After the backup, the partition 1 database is re-synchronized with partition 0, by applying all accumulated journal entries from partition 0 to partition 1.

This scenario provides for recovery of the production database onto the same partition or a different physical machine, with minimum inconvenience.

2.9.1.7 Consolidating worldwide operations

If a customer's business needs different time zones on a single AS/400 system, there are no AS/400 system values available to support them. For example, if a customer has a system located in Hong Kong and has offices located in the US, the users signing on to that system from the US offices will always have the Hong Kong time. The application has to manage the different time zones for users, which may be located around the whole world. Even if they can manage the different time zones by the application, they still need to find a way to manage their downtimes. Their system cannot be saved while the users in the US are still running their interactive applications. They need to have batch windows for running batch programs.

One way to provide different time zones to users is to split the AS/400 system from a single system image to a system with logical partitions.

2.9.1.8 Three-tiered application architecture

Several types of AS/400 applications, such as J.D. Edwards OneWorld, use a three-tiered architecture with a desktop PC, an Application Server and a back-end Database Server (see Chapter 5, "OneWorld implementation designs" on page 71). With the availability of AS/400 Logical Partitioning, customers can put the Database Server in the primary partition and Application Servers in secondary partitions within the same system. This can reduce footprint and maintenance costs. Furthermore, the Database Server can communicate with the Application Servers through the virtual OptiConnect function (see 20.2.3, "Virtual OptiConnect" on page 550) available between the partitions.

2.9.1.9 A perfect combination with Domino clustering

E-mail and groupware software, such as Lotus Domino for AS/400, becomes more and more critical in the daily business operation. Customers cannot afford to have their Domino servers fail for hours. This may affect customer service levels.

Lotus Domino for AS/400 has already taken advantage of the reliability and availability features of the AS/400 system, such as RAID-5, mirrored disk units, and integrated backup capability. With the AS/400 system's Logical Partitioning and Domino clustering, the level of availability is further enhanced.

Putting the Domino servers into two different AS/400 logical partitions isolates them from software errors affecting both servers. Plus, it also provides the ability to have high-speed cluster replication between the two AS/400 partitions using the virtual OptiConnect function.

2.10 Clustering

Logical partitioning creates independent systems within a single physical box. Clustering can be seen as a superset of Logical Partitioning in that it provides a single resource view that binds together two or more physical AS/400 systems. These can, in turn, be logically partitioned if required.

Clustering offers increased high availability of applications and data. It does this by providing automated recovery facilities across several combined AS/400 systems (defined as nodes in the cluster). For example, through clustering, an application can be set up to run on a primary node, with a secondary node defined for quick failover switching, when the primary node becomes unavailable. The switchover would include the automatic switching of communications to the secondary node.

It is also possible to imagine a cluster containing logically partitioned nodes. An application can run in one partition, in its own primary node, and have its backup secondary node ready on a logical partition in another node somewhere in the cluster. Figure 7 illustrates the concepts of clustering.

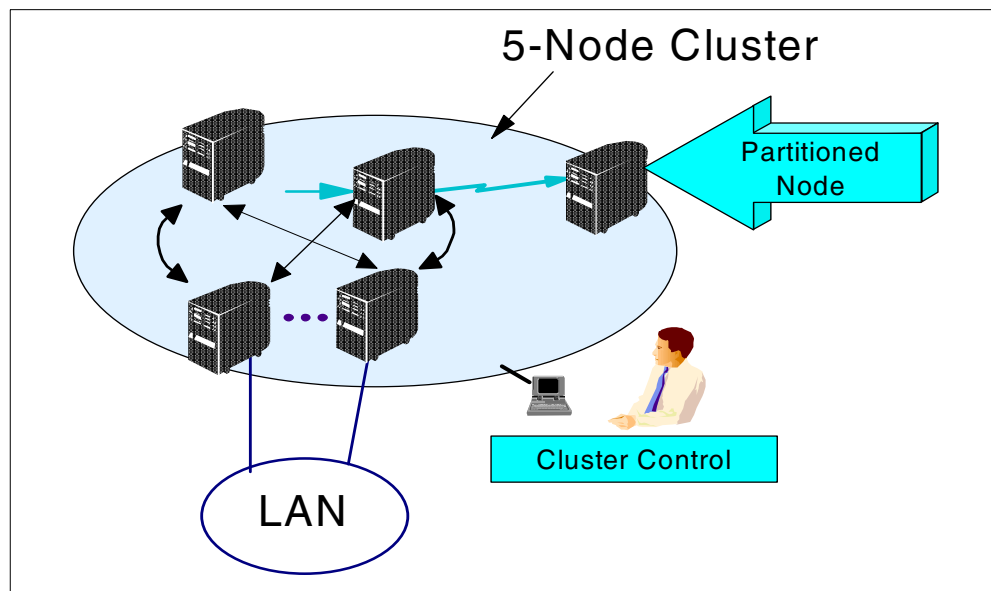


Figure 7. Clustered configuration with a partitioned node

2.11 System management

Integrated with OS/400 are a number of system management functions that are key to lowering the cost of ownership of the AS/400 system solution. These functions include:

- **Job management:** OS/400 includes an environment to manage jobs for users, operators, and programmers. An operator can look at all of the jobs running on the system, or select only those associated with a specific queue or user. An operator can sort them by CPU usage, view their properties, change their priority, stop them, or delete them within the limitations of authority granted to the operator. For more information on AS/400 jobs, refer to 7.1.5, "Jobs" on page 116.

- **Printer management:** The AS/400 system has a print output environment. An operator can select what output to see: either all of it by printer, user, or queue. With the printer management facilities, an operator can route the output to a different printer, change the number of copies, print only selected pages, or print duplex copies. AS/400 system print jobs can also be sent or mailed to other AS/400 systems and users. Plain text print output can be viewed on the AS/400 system before being printed. When using Advanced Function Printing (AFP), a user can see the actual AS/400 printed output (text, forms, or graphics) before it is printed. For more information, refer to Chapter 14, “Printing” on page 379.
- **Backup and recovery:** OS/400 has a backup scheduling application that allows an administrator to easily create a daily, weekly, and monthly backup plan along with other scheduled tasks. The administrator selects what to back up, where to back it up, and when to back it up. For more information, refer to Chapter 16, “Backup and recovery” on page 455.
- **User management:** When a new user is added to OS/400, besides entering the user ID and password, an administrator can specify the maximum amount of disk space the user can use to store information, the maximum priority at which any of their jobs run, and accounting information for tracking AS/400 system resource consumption by the user and several other attributes. The administrator can also specify the output queue and message queue for the user, the language and sort order for the user, and integrate the user ID and password with Windows NT, Novell NetWare, or OS/2 Warp Server for AS/400 running on the Integrated Netfinity Server (or Integrated PC Server) or with Domino for AS/400 running on the AS/400 natively.
- **National language support (NLS):** The AS/400 system has national language support for over 50 languages and can support multiple languages, sort sequences, and character sets per system. With this support, one user can interact with the system, for example, in the German language, while another user converses in French. Unicode (UCS-2) support is also available. For more information on national language support, refer to Chapter 17, “National language support” on page 471.
- **Problem management:** A complete problem management application is included with OS/400. This support allows administrators to track, analyze the causes of, prioritize, and report to IBM the problems associated with the computing environment. The AS/400 system keeps track of the installed products, release levels, and Program Temporary Fixes (PTFs) that were applied. For more information, refer to Chapter 22, “Problem identification and solution” on page 711.
- **Software fixes:** Individual software fixes or PTFs can be applied to LIC, OS/400, and licensed programs either permanently or temporarily. The benefit of applying a fix temporarily is that it can be applied, tested, and removed if it is appropriate to do so.

2.11.1 Electronic Customer Support (ECS)

ECS is a direct electronic link to IBM marketing, administration, technical, and service operations. ECS gives users and technical staff online advice and provides configuration management assistance, problem determination, and other service needs. ECS simplifies the PTF ordering procedure by enabling the speedy receipt and application of PTFs and quicker problem resolution. The

Copy-Screen facility permits a more accurate user-to-specialist problem description, and direct electronic contact with AS/400 system specialists. ECS support is provided for software problems (LIC, operating system, licensed programs and third party applications), as well as for hardware problems.

In addition to ECS, AS/400 system customers under warranty or an IBM maintenance agreement may receive Service Director support. Service Director monitors the AS/400 system in real time. When an error is entered into the problem log, it is immediately and automatically analyzed and reported to IBM through the ECS function on the AS/400 system. The problem is fed into the IBM remote technical assistance information network and IBM resources are used to evaluate the situation. If a PTF is available that corrects the problem, it can be downloaded to the AS/400 system. This can significantly reduce overall downtime.

In addition to these system management facilities, additional IBM and third-party products are available to manage larger and distributed AS/400 system environments. These products provide support for software distribution, client management, performance analysis, capacity planning, and more.

2.11.2 Management Central

A suite of system management functions known as Management Central includes system management functions such as collection services, object packaging, PTF management for distributed environment, and inventory on multiple systems. For more information on Management Central, refer to 11.2, “Operations Navigator and Management Central” on page 286.

2.11.3 System management facilities

A variety of tools and functions that provide system availability and management are available in the AS/400 system. Some are discussed in this section:

- **System Managed Access Path Protection (SMAPP):** SMAPP supports and automates the process of selecting which access paths should be protected. The system uses the EDTRCYAP value to estimate the amount of journaling to perform. The shorter the time that is in this value, the more journaling takes place, which impedes system performance, but it leads to shorter IPLs. The longer the value is, the longer IPLs are. However, the impact of journaling on CPU and DASD utilization is less.
- **Expert Cache:** Expert Cache provides a disk cache tuner option, which allows the AS/400 system to take advantage of available main storage capacity. It dynamically responds to system jobs to cache pages of data in main storage to reduce the time to process disk I/O.
- **Integrated hardware disk compression:** Beginning with OS/400 V4R3, the compression of data on disk is supported by OS/400. Data is dynamically compressed and uncompressed by the AS/400 DASD controller as data is written to and read from disk. Disk compression does not affect the main CPU utilization since this function is performed by the DASD controller IOP.

Support for Integrated Hardware Disk Compression is only provided by PCI DASD controllers (#2741, and #2748) and SPD DASD controllers (#6533 and #9754). In V4R3, the 17.54 GB drives are not supported. However, V4R4 of OS/400 adds support for compression on 17.54 GB drives. Compression is limited to user ASPs.

- **Hierarchical Storage Management (HSM):** OS/400 includes HSM APIs that are used by Backup and Recovery Media Services (BRMS), 5769-BR1, to provide HSM functions. These APIs can also be used to develop custom HSM applications. The APIs are documented in the *AS/400 Hierarchical Storage Management*, SG24-4450. Refer to the following Web site for more information on BRMS and HSM: <http://www.as400.ibm.com/hsmcomp>
- **PTFs available on Internet:** Beginning with V4R3, AS/400 customers can download PTFs over the Internet. The client hardware needed is a PC with Windows 95 or Windows NT, a TCP connection to the AS/400 system over a LAN, and access to the Internet. The various configurations and set up information are documented on the Web at:
<http://as400service.rochester.ibm.com>

Except for the medium of transport (Internet), the functionality is the same as the ECS method of transport. The user selects the PTFs and options using a Web browser and submits the order. At the referenced Web site above, the user can also search on PTF cover letters and read them before the order is even placed. The same entitlement rules that apply on the ECS connection are enforced. In other words, if a user can acquire PTFs electronically over the ECS, they can acquire PTFs over the Internet.

2.12 AS/400 advanced technologies

With the introduction of OS/400 V4R2, the AS/400 system offered important new capabilities in key areas such as Java, Web serving, Lotus Domino, and business intelligence solutions.

2.12.1 Java

Java is a key application development environment for the AS/400 system. As Java technology evolves from Sun, the AS/400 system takes advantage of the new functions and features of this environment.

The AS/400 Developer Kit for Java supports Sun's Java 2. With the concurrent Java support shipped as part of the V4R4 AS/400 Developer Kit for Java, you can install Java 2 on systems where JDK 1.1.6, 1.1.7, or 1.1.8 are already installed. A Java Virtual Machine (JVM), which resides *below* the Technology Independent Machine Interface (TIMI), therefore very close to hardware, enables fast interpretation and execution of Java code on the AS/400 system. JVM on the AS/400 also includes two types of compilers:

- **Just in Time Compiler (JIT):** Generates non-persistent 64-bit RISC machine instructions from Java bytecodes.
- **Class transformer:** Generates persistent machine instructions.

These two compilers enable the direct execution of Java on the AS/400 system without the overhead of interpretation.

High-performance garbage collection is provided by OS/400 to improve the performance and the scalability of Java. An advanced garbage collection algorithm allows Java to scale to the large numbers of objects expected when running enterprise applications on the server. As opposed to some other platforms, the AS/400 system does not require stopping all other applications and threads while running garbage collection.

OS/400's unique single-level-store architecture is also exploited to give Java objects on the AS/400 system an advantage that is not available on any other platform. Java objects on the AS/400 system can be full-fledged system objects that allow them to be persistent, shared, secure, backed up, and restored. This allows the AS/400 system to offer persistent Java objects with performance and support that is unparalleled in the industry. The AS/400 single-level-store technology permits Java objects to be stored in their object form without the performance and maintenance overhead of two-level-store operating systems.

Java applets and applications that access AS/400 programs and data from client workstations (or a Java-enabled server) can be written using the AS/400 Toolbox for Java. Java classes on the client can be used to access existing AS/400 applications and data using low-level APIs. This provides easy entry into Java development while leveraging what already exists on the AS/400 system today.

2.12.2 Web server for e-business

Typically e-business solutions for companies begin with posting information on the Web and connecting employees via e-mail. This grows into taking core business systems and extending their reach and value by enabling them for the Web. Eventually, employees, distributors, suppliers, and customers are all connecting with each other through Internet-based systems that let them dynamically share data and applications.

The HTTP server is at the core of all Web applications and services. The IBM HTTP Server for AS/400 is the foundation for AS/400 e-business solutions. The IBM HTTP Server for AS/400 combines the basic functions of a Web server with expanded functionality that allows for greater flexibility in establishing a Web presence.

OS/400 V4R4 provides support for Lightweight Directory Access Protocol (LDAP) in the HTTP Server for AS/400, which defines a protocol to access directory services on a network. A new Domino plug-in is also provided that allows the HTTP Server for AS/400 to access documents stored in Notes.

AS/400 Web serving capabilities include support for the IBM WebSphere family. There are several components in the WebSphere family:

- **IBM WebSphere Application Server:** Provides a framework for consistent, architected linkage between the HTTP requests and business data and logic. IBM WebSphere Application Server is intended for organizations that want to take advantage of the productivity, performance advantage, and portability that Java provides for dynamic Web sites.
- **IBM WebSphere Studio:** A set of PC-based tools to help developers create WebSphere applications.

IBM Net.Data allows the creation of interactive Web applications with "macros" to add logic, variables, program calls, and report writing to HTML. These macros combine the simplicity of HTML with the dynamic functionality of CGI programs, which makes it easy to add live data to static Web pages. Live data includes information stored in DB2 for AS/400 (locally or remotely), databases on other systems, REXX programs, C and C++ programs, programs in other AS/400 languages (such as CL, RPG, and COBOL), and other sources.

Web serving capabilities of the AS/400 system have also been extended with a powerful, full-text search engine through the implementation of NetQuestion in OS/400 Version 4 Release 3. NetQuestion provides the tools to build a centralized internet search service. NetQuestion can index both plain text and text with HTML markup and provides CGI scripts and HTML forms for searching and administration.

The AS/400 Web serving capabilities allow businesses, such as a shop, a service, or a distributor, to open an electronic storefront on the Web with WebSphere Commerce Suite V4.1 for AS/400. WebSphere Commerce Suite, which is a replacement for Net.Commerce, provides a complete e-commerce solution from catalog and storefront creation to payment processing and integration with back-end systems.

2.12.3 Lotus Domino

Lotus Domino is the world's leading workflow, messaging, groupware, and Web software. Lotus Domino enables you to communicate with colleagues, collaborate in teams, and coordinate strategic business processes on and off the Web.

Domino for AS/400 is the Lotus Domino server product running on a 64-bit AS/400 RISC processor. It requires OS/400 Version 4 Release 2 or later. Domino for AS/400 provides all the functionality of the Lotus Domino server that runs on other platforms, and more. Domino for AS/400 includes integration between Lotus Domino databases and DB2 UDB for AS/400 databases. Both real-time and scheduled integration of databases is available to meet a variety of application needs.

Automatic synchronization between the Domino Public Address Book and the AS/400 System Distribution Directory provides a powerful, integrated mail server for organizations with multiple e-mail products, including OfficeVision/400, POP3, JustMail, and Internet mail.

The Lotus Enterprise Integrator option to synchronize authorizations between DB2 UDB for AS/400 databases and Domino databases is a platform exclusive.

2.12.4 Business intelligence solutions

Business Intelligence (BI) is taking corporate data and turning it into decision support information. BI solutions have become much more affordable due to new innovations in software and hardware. One of these key technologies is data warehousing. Data warehouses provide the plumbing for BI applications and the advent of data warehouse technology and industry specific BI applications have made implementations meaningful and cost effective.

2.12.4.1 AS/400 enabling technology

The AS/400 system is the only hardware and software enabled for 64-bit relational database processing. This system has been optimized for a business intelligence environment with customized hardware (AS/400 servers), and optimized software (DB2 UDB for AS/400, SMP for DB2 UDB for AS/400, DB2 Multi-System, and Data Propagator Relational). These hardware and software functions combine to make a powerful business intelligence server, which is easy to install, manage, and use.

With the AS/400 system's open interfaces, hundreds of tools can be used to provide BI solutions accessing DB2 UDB for AS/400 data transparently. Such tools include desktop analysis tools, like Business Objects, and sophisticated multi-dimensional analysis (commonly referred to as OLAP) tools, such as Essbase/400, with no special programming required.

Symmetric Multiprocessing (SMP) for DB2 UDB for AS/400 provides parallel query processing. This allows multiple processors in a single AS/400 system to collectively work on a single query or one processor to work on multiple queries, which can improve query performance by as much as 400%. DB2 Multi-System support provides clustering for the AS/400 and allows up to 32 AS/400 systems to be “clustered” together into a single system. This clustering provides almost unlimited scalability and unparalleled performance for AS/400 customers.

Data replication is an important technology to facilitate the automated loading of data warehouses while cleaning up or summarizing data for integrity and performance purposes. DataPropagator/400 provides asynchronous data movement between OLTP systems and Business Intelligence systems. Data Propagator allows fields to be summarized, derived, or aggregated into the data elements necessary in your data warehouse.

Data mining is a BI application that utilizes mathematical algorithms to scan potentially large amounts of data to find the golden nuggets of information. Intelligent Miner for AS/400 provides the most advanced data mining application for AS/400 customers. Intelligent Miner provides advanced computer models to “discover” data relationships previously unknown. The models include algorithms for clustering, information classification, predictions, associations, sequential pattern recognition, and time sequence patterns. This analysis provides executives with insight that truly can be a competitive advantage.

2.12.4.2 Business Intelligence tools and applications

Virtually every major BI tool is supported on the AS/400 system. That includes tools for moving and cleansing data, such as DataMirror and ETI Extract. It also includes tools for organizing data in a multi-dimensional and relational format, such as Essbase/400 and DataTracker, and multi-dimensional analysis tools like Analyzer, Business Objects, and Cognos Powerplay. These tools allow customers unlimited flexibility in building their own Business Intelligence applications. They also allow applications to use AS/400 data and non-AS/400 data.

While there are many technical advantages for using the AS/400 system for your Business Intelligence server, the overwhelming reason why customers use the AS/400 system is the combination of its power and simplicity. The AS/400 system provides a full range of tools, applications, and hardware in single integrated platform, that help to make rapid implementation a reality. Large and small businesses alike agree. This is the ideal Business Intelligence server.

The AS/400 system provides outstanding database technology, which supports rapid access to large amounts of data. The AS/400 system supports a wide range of Business Intelligence solutions including small departmental applications and large Business Intelligence environments. The benefits of this application are measured by the better decisions that can be made as a result of having better information, and information in a format to support the decision-making processes of a company.

2.13 Why OneWorld on AS/400 system

Migration to client/server and network-centric computing models has suffered a number of high profile and expensive failures. The cost of building and maintaining complex hardware and applications has risen dramatically. With these facts in mind, the history of reliability, stability, and low maintenance costs of the IBM AS/400 become very attractive. Companies have come to realize that total cost of ownership, not just hardware costs, is what is really important with information systems.

OneWorld on the AS/400 system helps to solve many of these problems. The integrated operating system requires less setup and maintenance than other hardware configurations. The following list summarizes the reasons for implementing OneWorld on the AS/400 system:

- Reliability: AS/400 reliability is at 99.94%.
- Integration: The AS/400 Custom Server for OneWorld integrates database and Deployment Server into one solution.
- Powerful database that does not require a database administrator.
- Low cost of ownership.
- Speed to implement: ERP applications implement quickest on AS/400 system.
- Customer satisfaction: AS/400 has industry-leading customer satisfaction.
- Scalability: The AS/400 system can scale to effectively meet the needs of all sizes and shapes of J.D. Edwards customers.
- With Integrated Netfinity Server and LPAR the AS/400 system, OneWorld implementation can be simplified by reducing the number of boxes needed, and thus allowing customer to spend more time managing the business versus managing the network.
- J.D. Edwards WorldSoftware and OneWorld on AS/400 can co-exist on one system using a common database, DB2 UDB for AS/400. This allows customers a gradual migration.
- Education and dedicated support resources for successful implementation provided by IBM and J.D. Edwards.
- e-business options with AS/400 using the WebSphere set of products, Domino, and an array of third-party e-business solutions.

Chapter 3. OneWorld internal architecture

This chapter provides an overview of the OneWorld internal architecture. Basic knowledge of OneWorld internals is necessary if you work with troubleshooting and performance analysis. This chapter covers the following areas: OneWorld tool set, OneWorld application code creation and storing, and OneWorld application development flow.

3.1 OneWorld tool set

We start this chapter with a description of tools used to develop, maintain, and modify OneWorld.

Application development of OneWorld is accomplished by using a set of fourth generation language (4GL) tools. With the exception of C business functions, which are addressed later, applications are not written in third generation languages. These 4GL tools are used to create interactive applications as well as reports and batch update processes. The OneWorld tool set is included in OneWorld. Therefore, it can be used by J.D. Edwards programmers, business partners, and customers.

3.1.1 OneWorld design tools

OneWorld includes a number of 4GL design tools that help the developer design and create interactive and batch applications. Design tools generate specifications (Specs), which are later interpreted by the OneWorld runtime engines. Some of the design tools are listed here:

- **Table Design Aid (TDA):** Used to design tables and indexes for use by the applications.
- **Business View Design Aid (BDA):** Used to describe the data that will be presented to an application.
- **Interactive Application Design Aid:** Also known as Forms Design Aid (FDA), this is used to describe the appearance and functionality of a form that makes up interactive applications.
- **Batch Application Design Aid:** Also known as Report Design Aid (RDA), this is used to design the appearance and functionality of reports and batch update applications.
- **Business Function Design Aid:** A set of programs that helps standardize, catalog, and compile business functions written in the C language.

3.1.2 OneWorld runtime engines

OneWorld has two runtime engines. Interactive Runtime Engine handles interactive applications, while Batch Runtime Engine handles batch applications. The functions of the runtime engines include:

- Establish the environment and users for the session
- Perform security checks
- Read and cache control tables including the Object Configuration Manager (OCM)

- Establish I/O channels through OneWorld middleware
- Read and interpret the specifications created by the design tools for requested applications
- Interpret Event Rules for requested applications
- Call business functions

3.1.2.1 OneWorld Interactive Runtime Engine

The OneWorld Interactive Runtime Engine runs on a client workstation (or terminal server). It provides menu functionality as well as execution of interactive applications. All presentation logic required by the application is handled by this engine on the workstation. Business logic (Master business functions) can be handled by the workstation or passed off to a logic (application) server to be executed there.

3.1.2.2 OneWorld Batch Runtime Engine

The OneWorld Batch Engine is also known as Universal Batch Engine (UBE). It is designed to handle the printing of reports and to perform batch file updates. Batch application Specs created by RDA are read and interpreted within the engine.

There are actually two versions of this engine: one that runs on the client workstation and one that runs on the logic server. The server version of the UBE engine performs batch processing through kernel processes. The number of kernels can be adjusted to obtain maximum performance for the specific server.

Attention

UBEs are CPU intensive jobs. In many cases, executing a UBE will completely occupy one CPU on the logic server.

3.1.3 OneWorld subsystems

In an attempt to better use multiprocessor machines and reduce batch processing times, J.D. Edwards developed Parallel Batch Processing, which is also known as OneWorld Subsystem Processing. In this context, parallel batch processing refers to the following tasks:

1. Divide the data into separate, independent streams.
2. Run the original program independently against each stream.
3. Combine the results back together.
4. Evaluate the report process flow to eliminate redundancies.

A OneWorld subsystem (not to be confused with an AS/400 subsystem) is a process that runs continuously, similar to a Windows NT service or a UNIX daemon. It is set up to aid in the execution of a specific batch process. A subsystem report, once started, will loop continuously while reading records from the subsystem table, F986113, until it is explicitly stopped.

The advantages of subsystem processing are:

- Simplifies report execution (eliminates the need to specify data selection, data sequencing, and processing option values).

- Allows multiple runs of the same report without overhead of reloading report specifications each time. This is especially significant for small, simple reports where almost all of the report processing time is spent in initialization (creating an environment, loading specifications, etc.).
- Enables the elimination of redundant logic by processing it once per subsystem, instead of once per report run.
- Allows parallel processing because all subsystem reports read from the same table (F986113).

Figure 8 shows a comparison of process flows with and without the UBE subsystem.

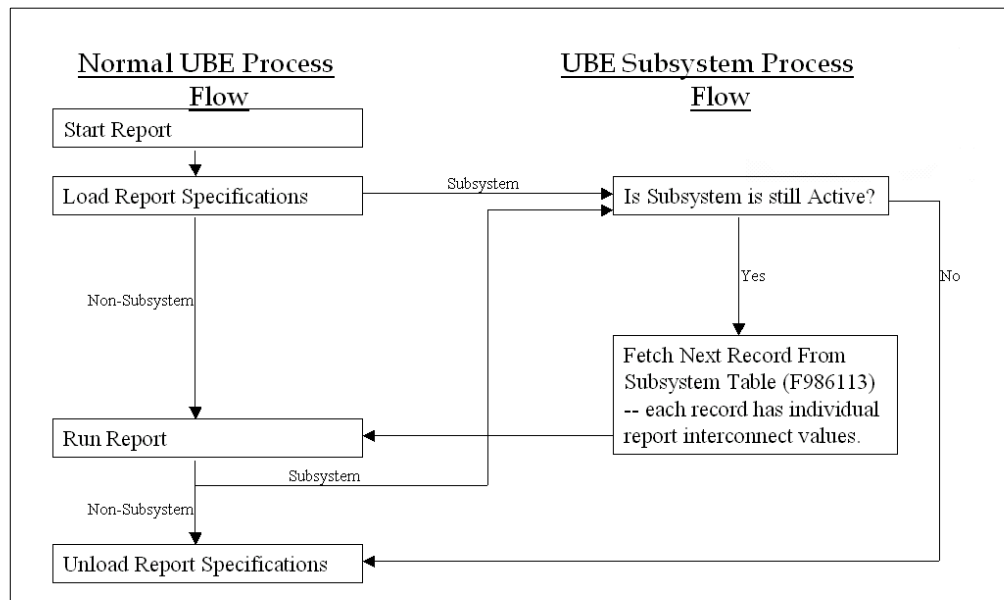


Figure 8. Subsystem UBE process flow

3.1.4 Update process for OneWorld tools

The OneWorld Design tools, runtime engines, and middleware are updated through Service Packs. Each release of the applications include a specific Service Pack. For example, Release B7332 of the applications comes with Service Pack 9. During a fresh installation, both the applications and service packs are installed together. After that, they can be installed independently of each other.

3.2 OneWorld application code creation and storage

To understand the internal operations of OneWorld, it is necessary to look at the components of an application and how they are stored.

3.2.1 OneWorld application specifications: Building blocks

As expected, the building blocks of a OneWorld application are created by the OneWorld design tools. These components or objects are cataloged in the Object Librarian for easy tracking. Figure 9 on page 34 shows a flow chart of how the application building blocks fit together.

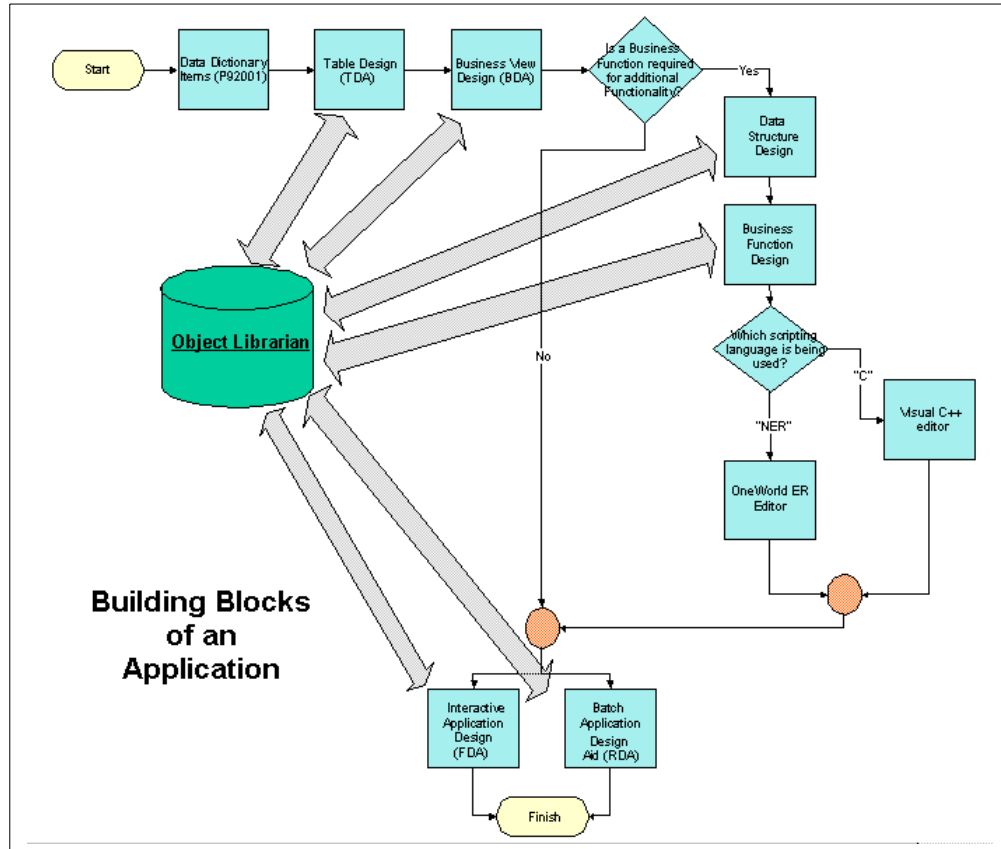


Figure 9. Application building blocks

3.2.2 Specification storing: Central objects

The application specifications (Specs) are held in two different forms: runtime form and relational database form.

The runtime form, also known as Table Access Management (TAM) Specs, is a format that is easy for C language functions to read. The term TAM refers to a J.D. Edwards proprietary single-user indexed file system that is used on the client (and logic servers) to deliver the specifications to the runtime engines during execution. When the AS/400 system is used as a logic server, the TAM files are stored in the integrated file server (IFS).

The storage form of the application specifications is known as Relational Database (RDB) Specs, which are also known as *central objects*. In this format, a relational database is used to store the specifications in binary form. Central objects on the AS/400 system were supported as a special Service Pack in Release B733.1 and are included in standard Release B733.2 and later releases.

Before they can be used by the runtime engines, the RDB Specs must be converted into their TAM format. This process normally happens for all applications during the package build process. During application development, a single application can be converted through the *check out process*. The *check in process* converts the TAM Specs back into their RDB format and stores them in the central objects.

Figure 10 shows the flow of the components during application development.

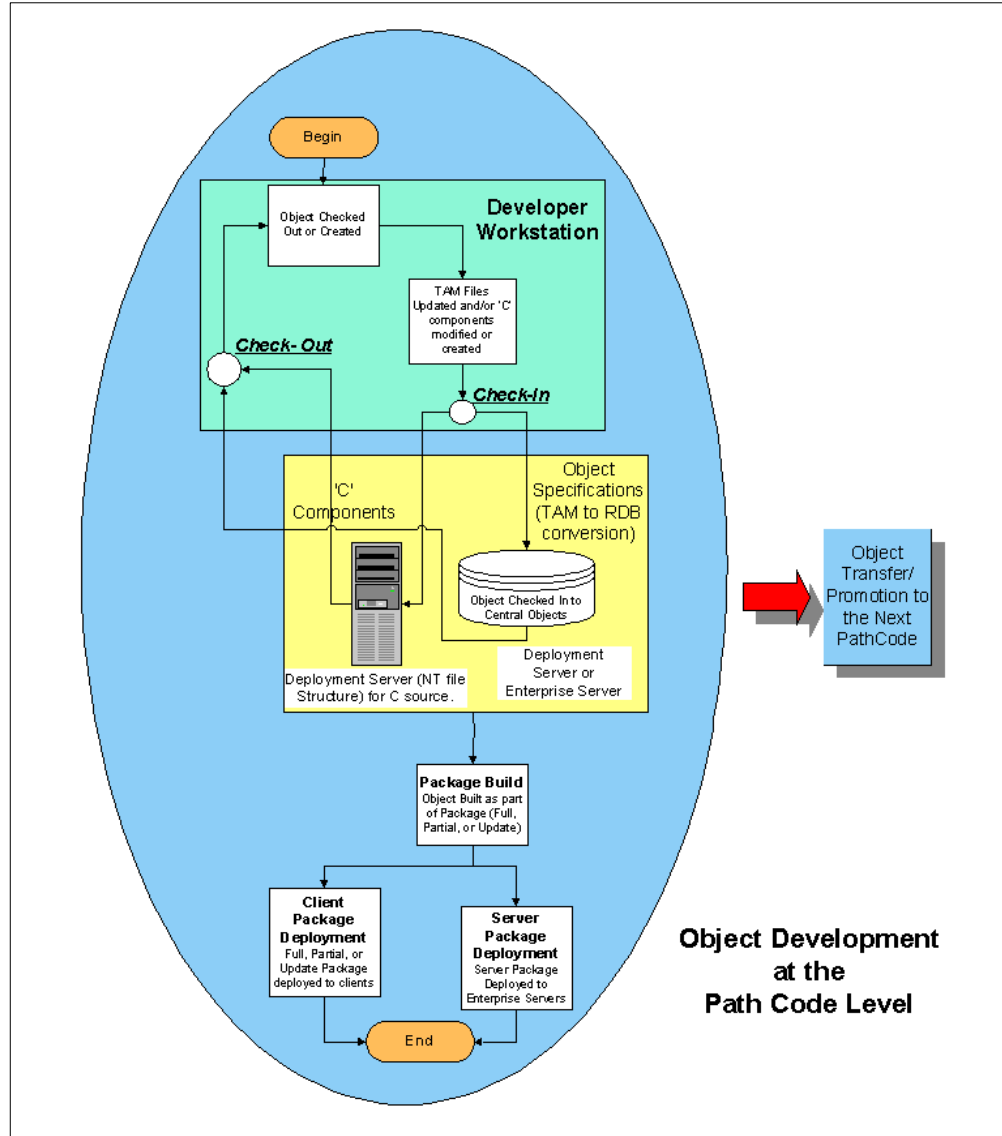


Figure 10. Application development flow

3.2.2.1 Check in and check out process

Object check in is the process of copying TAM specifications or C components from a workstation to two central locations:

- Central objects for TAM Specs
- The Deployment Server for C components

During check in, TAM Specs are taken from the developer's workstation and then converted into RDB Specs to be placed into central objects. C components are taken from the developer's workstation and copied into the Deployment Server Windows NT directory structure (under the proper path code).

The process described above is reversed for check out.

Figure 11 shows what happens at check in or check out for any object in OneWorld (UBE versions are slightly different).

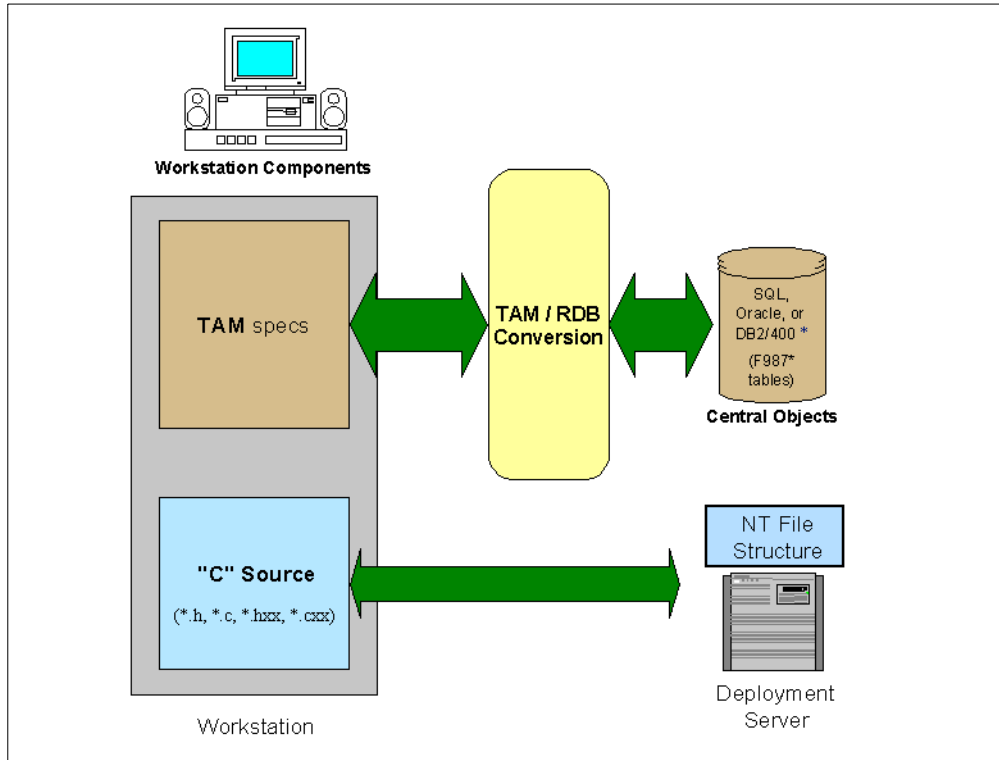


Figure 11. Check in and check out process

3.2.3 Events and Event Rules

The OneWorld tool set creates “event driven” applications. This means that during the application development process, the ability exists to attach functionality when certain occurrences are detected. Examples of events in an interactive application may be the exiting of a control or a button being pressed. Examples of events in a batch application may be a control break being detected or the end of a table being reached.

Event Rules (also known as Embedded Event Rules) are used to extend the basic behavior of an interactive application or a batch application. Examples of tasks that can be performed by Event Rules are:

- Perform mathematical calculations
- Perform table I/O
- Hide or display controls
- create and use variables
- Initiate workflow
- Call business functions

Event Rules, which are written in a scripting language, are included inside the application. These rules are read and interpreted by the runtime engines.

If a specific group of event rules are used in more than one application, they can be converted from Embedded Event Rules into Named Event Rules and compiled.

3.2.4 OneWorld business functions

A OneWorld business function is a compiled object that carries out a particular function. For example, this business function could edit a particular field, perform validation against a database table, or return a computed value to the calling program or business function.

OneWorld business functions are written in two languages: C and Named Event Rules (J.D. Edwards proprietary language).

3.2.4.1 Named Event Rules

Named Event Rules (NER) are business functions that are created using the J.D. Edwards Event Rules editor. Functions are coded and written using Event Rules and are then compiled. During compilation, the equivalent C source is generated. From this point on, the NER object is treated the same as a C business function.

Business functions written in the NER language have come close to functionality provided by C language functions, but still lack some functionality that has to be written using C as the source language.

3.2.4.2 C language business functions

C business functions are created using the OneWorld business function Design Aid and the Visual C++ editor. Functions are coded and written using the C source language and syntax. The developer must be proficient with C and know the coding rules, syntax, and constraints. Even though C business functions are written using the Visual C++ editor, they have to be compiled using the OneWorld BusBuild compiler.

3.2.5 Update process for applications

For more information on updates and ESUs, refer to 9.2, “OneWorld software maintenance” on page 243.

3.3 OneWorld application development flow

The standard installation process for OneWorld includes a specific development methodology. It is designed for the easy creation of separate (and isolated) OneWorld environments to allow for application development, testing, Conference Room Pilot, and production. A OneWorld environment is the combination of Specs, executables, control tables, and business data that are used in a specific OneWorld session.

The term *path code* needs to be mentioned here. The path code determines the set of Specifications and executables that will be used by an environment when an application is run. A path code can be used by one or more OneWorld environments. Refer to 4.3, “Path codes” on page 52.

3.3.1 OneWorld standard environments and path codes

The installation creates several environments by default. These environments are:

- Development (DEV)
- Test (TST)

- Conference Room Pilot (CRP)
- Production (PRD)

The purpose of these phases is to ensure the application integrity, as applications are created, tested, changed, and finally moved into the production environment. Here is a description of each environment:

- **Development:** A place where changes are keyed and given the first tests by the programmer.
- **Test:** An environment for the Quality Assurance Department to test groups of changes together using Development's test data and a new path code including a group of transferred changes.
- **Conference Room Pilot (CRP):** An environment for the actual users of the software (Accounts Receivable for example) to run the applications against real-life data to determine if the needed functionality is present.
- **Production:** An environment where actual work is being done and the business is being run.

Figure 12 shows this methodology.

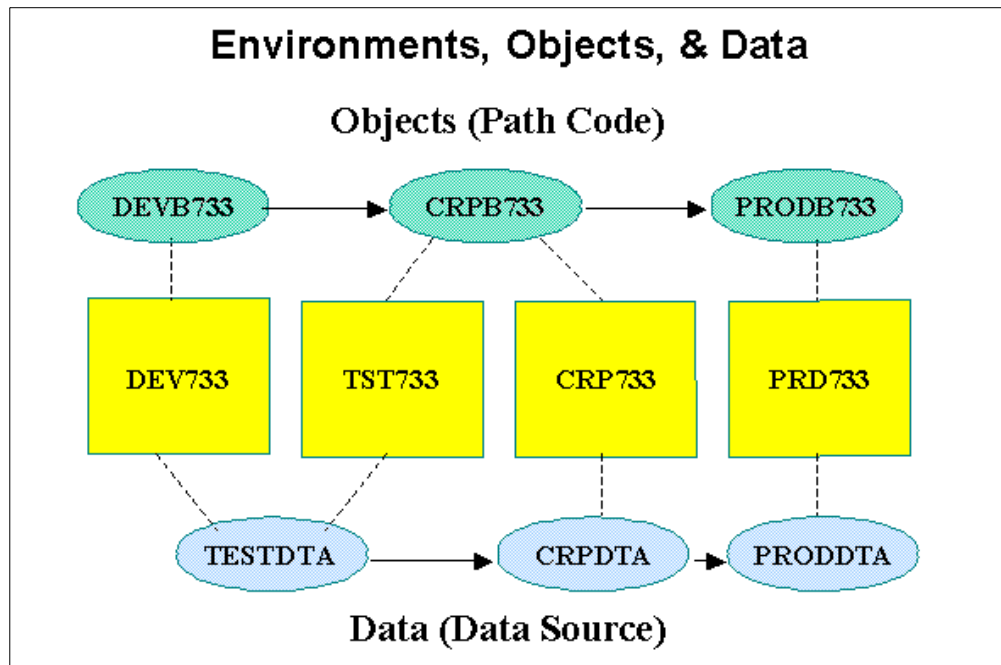


Figure 12. Standard environments and path codes

3.3.2 OneWorld objects

OneWorld uses an object storage management concept similar to personal computers and UNIX operating systems. OneWorld implementation on the AS/400 system uses an integrated file system (IFS) to store information about the OneWorld software environment. The AS/400 library structure is used to store executable programs and the database.

IFS directories are used to store OneWorld stream file application objects. IFS is also used to access the J.D. Edwards OneWorld program library (<pgm_lib>) in QSYS.LIB using pointers called *symbolic links*. A symbolic link, which is also

called a *soft link*, is a path name. When the system encounters a symbolic link, it follows the path name provided by the symbolic link and continues on any remaining path that follows the symbolic link. If the path name begins with a /, the system returns to the "/" (root) directory and begins following the path from that point. If the path name does not begin with a "/", the system continues searching in the same directory as the symbolic link.

Unlike a hard link, a symbolic link is an object (of object type *SYMLNK). It can exist without pointing to an object that exists. You can use a symbolic link, for example, to provide a path to a file that will be added or replaced later. Also, unlike a hard link, a symbolic link can cross file systems. For example, a user of a program that is working in one file system can use a symbolic link to access a file in another file system.

For further information, refer to *Integrated File System Introduction*, SC41-5711.

3.3.2.1 OneWorld libraries

The libraries in Table 1 are associated with the OneWorld, and are created during a *typical* OneWorld installation process. AS/400 libraries are created automatically in a task that follows Deployment Server installation. Depending on your particular installation, you may not have all of these libraries on your system.

Table 1. J.D. Edwards OneWorld installed libraries

Library name	Library type	Description
B7332SYS	System	Contains the commands and programs for administering OneWorld on the AS/400 system. The default OneWorld data source name is System - B733.
B733MAP	System	Contains the server map tables (F986101, F98611, F986110) that are used by the logic server. The default OneWorld data source name is <host_name> - Server Map.
CLTCOM	Coexistence control tables	Contains common User Defined Codes (UDCs) and Menus tables for both OneWorld and World. The default OneWorld data source names are AS/400 COMMON, AS/400 COMMON - CRP and AS/400 COMMON - Production.
CLTDTA	Coexistence production	Contains common production data for both OneWorld and World. The default OneWorld data source name is Business Data - PROD.
COCRPB733	CRP	AS/400 OneWorld Central Objects for the CRP environment. The default OneWorld data source name is Central Objects - CRPB733.
CODEVB733	Development	AS/400 OneWorld Central Objects for the Development environment. The default OneWorld data source name is Central Objects - DEVB733.
COPRDB733	Production	AS/400 OneWorld Central Objects for the Production environment. The default OneWorld data source name is Central Objects - PRODB733.
COPRTB733	Pristine	AS/400 OneWorld Central Objects for the Pristine environment. The default OneWorld data source name is Central Objects - PRISTB733.

Library name	Library type	Description
CRPB733	CRP	OneWorld C programs and business functions compiled on the AS/400 system for execution in the CRP environment on the AS/400 system.
CRPB733FA	CRP	OneWorld full package library for the CRP environment, containing OneWorld C programs and business functions compiled on the AS/400 system, for deploying to the AS/400 Enterprise Server.
CRPB733DNT	Versions	Versions Tables (F983051 and F98306) for the CRP path code. The Version - CRPB733 - DNT data source points to the F983051 table. The Version - CRPB733 data source points to the F98306 table.
CRPCTL	Control	Menu tables for the OneWorld CRP environment. The default OneWorld data source name is Control Tables - CRP.
CRPDTA	Business Data	Conference room pilot (CRP) data for OneWorld and WorldSoftware. The default OneWorld data source name is Business Data - CRP.
DDB733	Data Dictionary	Data Dictionary tables for OneWorld. The default OneWorld Data Source name is Data Dictionary - B733. This data source is created during installation.
DEVB733	Development	OneWorld C programs and business functions compiled on the AS/400 system, for execution in the Development environment on the AS/400 system.
DEVB733DNT	Versions	Versions Tables (F983051 and F98306) for DEVB733 Path Code for the Pristine path code. The Version - DEVB733 - DNT data source points to the F983051 table. The Version - DEVB733 data source points to the F98306 table.
DEVB733FA	Development	OneWorld full package library for the Development environment, containing OneWorld C programs and business functions compiled on the AS/400 system, for deploying to the AS/400 Enterprise Server.
JDEOW	System	OneWorld Installation Library. Contains the AS/400 menus, commands, programs, and files required to install OneWorld on the AS/400 Enterprise Server.
OLB733	Object Librarian	Object Librarian tables. The default OneWorld data source name is Object Librarian - B733.
OWJRNL	System	Contains the AS/400 objects used for OneWorld journaling.
PRDB733DNT	Versions	Versions Tables (F983051 and F98306) for the Production path code. The Version - PRODB733 - DNT data source points to the F983051 table. The Versions - PRODB733 data source points to the F98306 table.

Library name	Library type	Description
PRISTB733	Pristine	OneWorld C programs and business functions compiled on the AS/400 system, for execution in the Pristine environment on the AS/400 system.
PRISTDTA	Pristine	Pristine environment data tables. Pristine data is different from the demonstration data in JDFDATA. The default OneWorld data source name is Business Data - JDE.
PRODB733	Production	Production environment library. Contains OneWorld C programs and business functions compiled on the AS/400, for execution in the Production environment on the AS/400 system.
PRODB733FA	Production	OneWorld full package library for the Production environment, containing OneWorld C programs and business functions compiled on the AS/400 system, for deploying to the AS/400 Enterprise Server.
PRODCTL	Control	Production environment control tables. Contains menu tables for the OneWorld production environment. The default OneWorld data source name is Control Tables - Production.
PRODDTA	Production	Production data for both OneWorld and WorldSoftware. The default OneWorld data source name is Business Data - PROD.
PRSB733DNT	Versions	Versions versions tables (F983051 and F98306) for the Pristine path code. The Version - PRISTB733 - DNT data source points to the F983051 table. The Version - PRISTB733 data source points to the F98306 table.
PRSTB733FA	Pristine	OneWorld full package library for the Pristine environment, containing OneWorld C programs and business functions compiled on the AS/400 system, for deploying to the AS/400 Enterprise Server.
QCPA	IBM supplied	IBM library that contains JDE objects.
QGPL	IBM supplied	IBM library that contains JDE objects.
SYSB733	System	OneWorld System tables. The default OneWorld data source name is System - B733.
TESTCTL	Control	Test environment control tables. Contains menu tables for the OneWorld test environment. The default OneWorld data source name is Control Tables - Test.
TESTDTA	Business Data	Test environment data tables. Contains test data for both OneWorld and WorldSoftware. The default OneWorld data source name is Business Data - TEST.
<p>Note: Integrated file system (IFS) objects do not get backed up using a SAVLIB command. Instead, the SAV command must be used. For more information, refer to the "Saving Objects in Directories" section of <i>Backup and Recovery</i>, SC41-5304.</p>		

3.3.2.2 OneWorld system library

Before starting OneWorld, you should add the OneWorld system library to the top of your library list. In the examples in this chapter, the name of the OneWorld system library is B7332SYS. The system library B7332SYS contains the following objects:

- **INI**: Identifies the jde.ini file used to initialize OneWorld on the AS/400 Enterprise Server.
- ***PGM** and ***SRVPGM**: The various programs and service programs required to run the OneWorld AS/400 Enterprise Server.
- ***MODULE**: The objects used to create OneWorld database programs that OneWorld uses to interface with DB2 UDB for AS/400.
- **H (*FILE)**: The file containing OneWorld header file members required to build new objects on the OneWorld AS/400 Enterprise Server.
- **JDE_HP4PS (*WSCST)**: The workstation customization file used to transform the EBCDIC PostScript file to ASCII for auto-detecting PostScript printers.
- **CHGLIBOWN (*CMD)**: A OneWorld utility command used to change ownership of all objects contained in a library.
- **SHOW (*CMD)**: A OneWorld utility command used to show runtime output.
- **UPDLF (*CMD)**: A OneWorld utility command used to modify the maintenance attribute of logical files. This applies to coexistence environments.
- **DSPSTMF (*CMD)**: Display Stream File is used to display AS/400 integrated file system (IFS) text stream files.
- **JDE.LOG** and **JDEDEBUG.LOG**: OneWorld log files that typically reside in a directory called JDE *release*, where release represents the OneWorld release, for example, /JDEB733.
- **LINKBSFN (*CMD)**: A command used to re-link business functions into their respective service programs (*SRVPGM). Typically, the system uses this command during an upgrade of the OneWorld system library.
- **PID2JOB (*CMD)**: The Convert Process ID to Job command returns the job information when the system passes a process ID to the command. The system writes the process ID in the JDE.LOG files. This command only returns job information while the job is still active.
- **PORTTEST (*CMD)**: This command runs the OneWorld test program PORTTEST.
- **RUNUBE (*CMD)**: Use this command to interactively run a UBE. If you need to run a batch UBE, use the SBMJOB command to submit the RUNUBE command to batch.
- **SAW (*CMD)**: Use this command to start the Server Administration Workbench.
- **QXTSRV (*FILE)**: The file containing the source for the workstation customization object for auto-detecting PostScript printers.
- **PRINTQUEUE (*FILE)**: The file containing the UBE output. This output is stored as ASCII PDF members.
- ***PGM** and ***SRVPGM**: The programs and server programs required to run the OneWorld network.

- **JDENET (*JOBQ)**: The job queue used by the OneWorld AS/400 Enterprise Server to service OneWorld workstations.
- **NETJOB (*JOBQ)**: The job description used by OneWorld AS/400 network jobs.
- **DBMONJOB (*JOBQ)**: The job description used by the database to monitor jobs.
- **JDENET (*CLS)**: The class used to create the routing entry for the JDENET subsystem.
- **ENDNET (*CMD)**: The command to end the OneWorld AS/400 network jobs and clean up the network runtime structures.
- **IPCS (*CMD)**: The utility command used to determine the status of objects used by the OneWorld AS/400 network jobs and as a backup method for cleaning up the IPCS objects.
- **STRNET (*CMD)**: The command to start the OneWorld AS/400 network jobs.

The QCPA IBM library can also be added to the top of your library list. The IBM library QCPA contains the Common Programming API support.

OneWorld directories in the integrated file system

AS/400 directories are created automatically in a task that follows Deployment Server installation, as well as while running of the installation workbench. The following list provides additional information about each directory:

- **B7332SYS**: Contains the OneWorld kernel specification files for the AS/400 system.
- **CRPB733**: Contains OneWorld specification files for the CRP environment.
- **DEVB733**: Contains OneWorld specification files for the Development environment.
- **JDEB7332**: Contains jde.log and jddebug.log files for OneWorld jobs running on the AS/400 system.
- **OneWorld**: Contains a subdirectory for each OneWorld package. Each of these package subdirectories contain the compiled OneWorld specifications for that package.
- **PRISTB733**: Contains OneWorld specification files for the Pristine environment.
- **PRODB733**: Contains OneWorld specification files for the Production environment.
- **TESTB733**: Test directory.

3.3.3 Client and server packages

A *package* is the term that describes the usual form in which groups of changes are delivered. A package includes the specifications and business function executables that will be used by a client or logic server. A standard release of OneWorld, B733.2 for example, comes with a pre-built package for the clients and each type of server. For any changes from the standard package, a package build needs to be performed. Many packages can exist at any one time for an installation. Packages can be activated and deactivated at different points in time.

For example, the Conference Room Pilot may not be using the same package as the Production environment.

Figure 13 and Figure 14 show the process by which workstation (client) and server packages are created. For more information, refer to on packages, refer to Chapter 13, “OneWorld package management” on page 327.

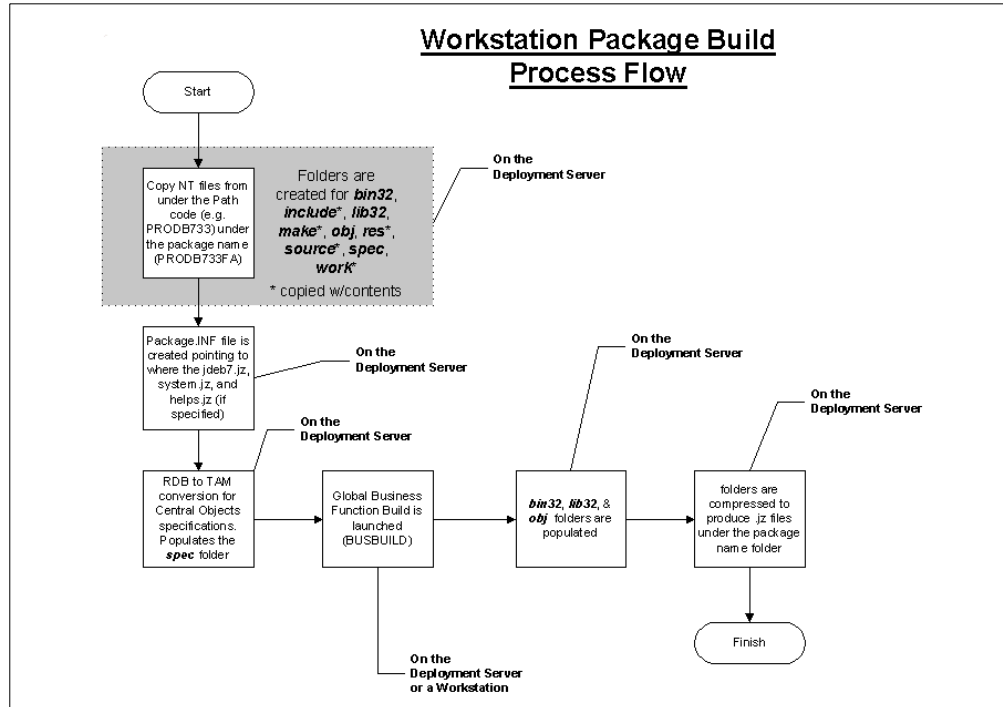


Figure 13. Workstation (client) package build

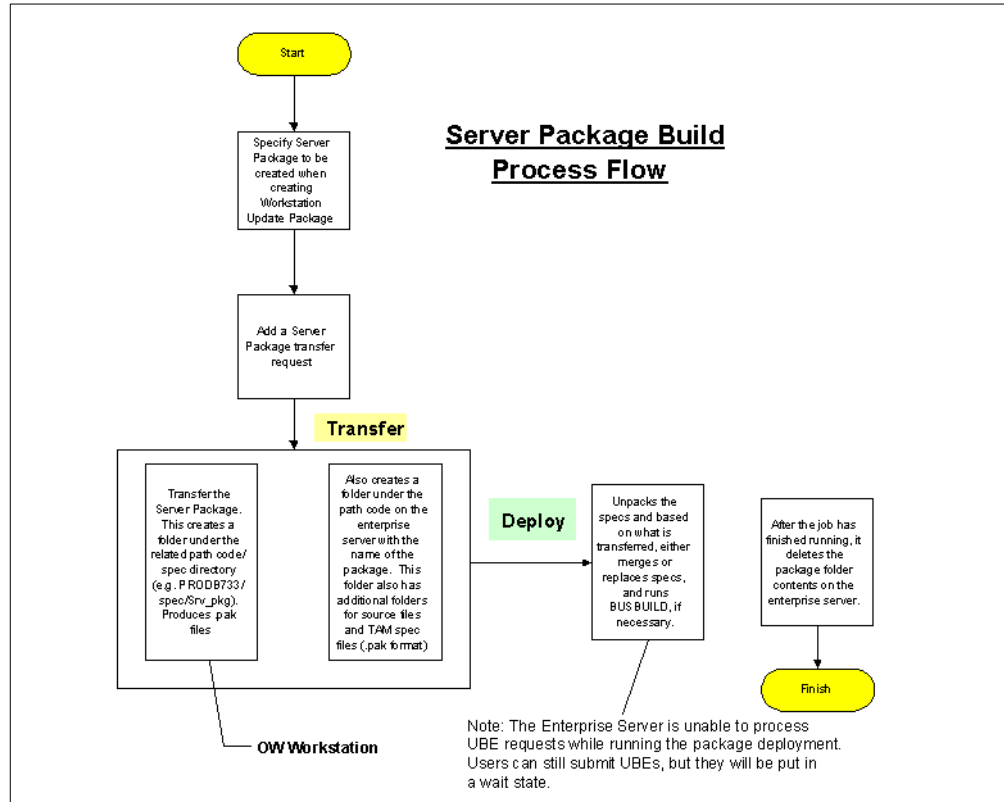


Figure 14. Server package build

Chapter 4. Configurable Network Computing (CNC)

Configurable Network Computing (CNC) is a term coined by J.D. Edwards to describe their network-centric software and hardware architecture. CNC allows highly configurable, distributed applications to run on a variety of platforms without users needing to know which platforms or which databases are involved in any given task. CNC insulates the business solution from the underlying technology, allowing enterprises to grow and adopt new technologies without having to rewrite applications.

4.1 Definition

Configurable Network Computing is an application architecture that allows interactive and batch applications, composed of a single code base, to run across a TCP/IP network of multiple server platforms and selected database management systems. The applications consist of reusable business functions and associated data that can be configured across the network dynamically.

OneWorld consists of the following components:

- **Design tools:** OneWorld provides a unified set of tools to create all interactive applications, batch applications, and reports.
- **Applications:** OneWorld provides the interactive and batch applications that perform your business needs. For example, Purchase Order Entry and General Ledger Post are applications.
- **OneWorld foundation code:** OneWorld provides underlying core processing that both interactive and batch applications depend on to run.
- **OneWorld middleware:** OneWorld provides middleware that insulates your applications from the underlying database, operating system, hardware, messaging systems, and telecommunications protocols. This insulates your business solution from the platform technology.

4.1.1 Benefits

The OneWorld Configurable Network Computing architecture provides the following advantages, which are explained in more detail later in this section:

- Network-centric software
- Flexible, leveraged technology
- Support for worldwide business
- Custom solutions without consequences

4.1.1.1 Network-centric software

Network-centric software allows you to create a uniform interface that supports a multi-platform network. This compatibility across platforms provides:

- Immediate availability of enhancements to all supported applications. Changes to the following enhancements are reflected in applications across the network:
 - Business objects
 - Business rules
 - Modes of processing
 - Hardware and database

- Platform-neutral business specifications, or middleware, that comprise a common set of APIs, which integrate multivendor and multiprotocol differences. This integration insulates developers from the need to understand or program to a specific platform.
- Support for Internet technology, such as a Web browser interface.

4.1.1.2 Flexible, leveraged technology

You create your applications using tools that do not require a designer to master a programming language. OneWorld tools conceal the code and allow the designer to concentrate on creating applications specific to the current business needs and accommodate changes to business rules without reprogramming the application source code.

OneWorld is object-based and event-driven to provide you with more efficient business processes. Developers can reuse objects between applications for different purposes. This reusability provides a consistency throughout all OneWorld applications.

OneWorld does not rely on one command or keystroke to process information, but rather processes information at strategic moments during the use of an application. For example, when a user moves between fields on any given form, the system processes the information at the moment the cursor leaves the field. OneWorld immediately notes any errors and hides processing, such as an update of files that may also store information for the field, when the user moves to the next field on a form.

In addition, OneWorld provides a common interface between applications. When you move from form to form, you see the same general setup.

4.1.1.3 Support for worldwide business requirements

You can run OneWorld on different platforms, including servers, PCs, or even laptops. This scalability allows a travelling consultant to work with OneWorld while traveling. OneWorld also provides support for mixed currency and languages.

If you already use WorldSoftware, you do not need to completely change your system to OneWorld. WorldSoftware and OneWorld can coexist by accessing the same database and supporting the same business task. Your business can continue its business practices without conforming to a single technological direction. You preserve your existing investment in technology, without affecting end-user productivity while you explore new business solutions.

4.1.1.4 Custom solutions without consequences

You can make custom solutions to business applications with little or no consequences when you upgrade to a new release of OneWorld. In other words, the OneWorld toolset acts as an “idea-enabler” by allowing you to transform a concept into a viable business solution. You maintain consistency across your enterprise, retain the flexibility to adapt to changing business requirements, and minimize the time required to implement upgrades. Examples of areas in OneWorld that you can customize without consequences during an upgrade include:

- Vocabulary overrides
- User overrides

- Versions
- Processing options
- Code generator options

4.1.2 Recommendations

Distributed processing involves the configuration of executable application components in various modes of operations and the configuration of data tables that the application components require. With OneWorld, you can dynamically configure application components and data tables. This process allows you the most flexibility during the configuration of your enterprise.

The key to achieving a high level of performance and seemingly boundless scalability is OneWorld's dynamic configuration of both hardware and software. However, the extensive amount of flexibility and options can sometimes appear contradictory and confusing. Consider the following guidelines for Configurable Network Computing:

- **Fewer is better:** A thirty-computer network is much more difficult to manage than a single computer. Although it may be necessary at times to add computers to your network, try to use the fewest computers necessary.
- **Homogeneity:** A homogeneous network is preferred to a heterogeneous network. Although OneWorld supports a heterogeneous network, avoid heterogeneity unless it is absolutely necessary. However, as time passes, a mixed hardware and operating environment is almost unavoidable, and OneWorld and CNC will support it.
- **Batch processing has some advantages:** In general, batch processing is more computer efficient than interactive or real-time processing. For example, electronic commerce or electronic data interchange (EDI) is a superb way to process transactions, and it keeps life simple.

Store-and-forward processing is another form of batch processing. It offers most of the advantages of real-time editing and allows quick response time at workstations.

- **Data warehousing:** Off-loading most reports to a secondary computer or computers offers the following benefits:
 - Relieves the processing burden from the main server and protects the response time of interactive users
 - Offers the full power of the computer to the report writer function, which provides faster processing
 - Allows you to restructure and simplify data for ease of use
 - Allows you to supplement or enhance data with exterior database information
 - Allows you to analyze data with modern analysis tools such as online analytical processors (OLAP)

Fifty to seventy percent of computer processing cycles are used for reporting and data analysis. Off-loading this processing to a separate data warehouse provides a substantial opportunity to ensure satisfactory performance for both the interactive and batch users.

- **Processing mode:** You should use real-time processing for low-volume data entry only. Use store-and-forward processing for high-volume data entry. You

can use the batch-of-one concept (asynchronous processing) as a compromise between real-time and store-and-forward processing.

You can perform data-entry functions, such as accounts payable vouchers, payroll time sheets, employee expense reports, and cash receipts, in a store-and-forward mode rather than a real-time mode to minimize the impact on the main server computer.

- **Multi-tiered networks:** Third-party database management systems consume a great deal of computer resources. Separating the J.D. Edwards applications and all their business objects onto a separate computer helps ease the burden on a computer. Splitting these two functions on two different computers allows optimum computing power for both needs. This implies a three-tier network. For example, such a network may consist of hundreds of workstations that direct into dozens of Application Servers that direct into a few database servers. The data warehouse acting on top of this would create yet a fourth tier. These techniques of multi-tiered distribution of work, while complex, offer a considerable opportunity to improve performance and increase scalability.
- **Multiple servers by vertical:** OneWorld can work concurrently with mixed databases across multiple machines. You can access business objects and data from several databases including IBM DB2 UDB for AS/400, Oracle 7 Server, Microsoft SQL Server, and the Microsoft Access. You can spread this data over several computers. For example, you can have separate servers for manufacturing and logistics, accounts receivable, and the general ledger and financial applications. You do not need a single central server. The concept of multiple servers working in harmony is central to the concept of CNC.
- **Data replication:** J.D. Edwards offers several facilities for data replication. These include:
 - OneWorld application for data replication
 - Just-in-time replication
 - Batch application

In addition, OneWorld can cooperate with several other systems tools that support the publishing and replication of data. Although some circumstances may arise when you need to perform data replication, such as store-and-forward processing, you should generally discourage the activity due to the difficulty in coordinating the replication successfully.

- **Departmental and application work groups:** Data replication is simpler when you use fewer computers. For this reason, it is often preferable to replicate data to an Application Server in a single group that performs the same business functions, such as credit and collections, accounts payable, sales order processing, and so on. In a departmental work group, workstations would have no replicated data. The business objects alone would be replicated and maintained on the departmental Application Server. This setup minimizes the data transfer between the workstation and the Application Server thus decreasing the amount of data replication. This configuration represents a classic three-tier architecture.
- **Heavy or light workstation processing:** You can configure either a heavy or a light workstation depending on your configuration of business objects, data, and mode of operation. The decision is entirely up to you.

4.2 Environments

A OneWorld environment consists of Object Configuration Manager (OCM) mappings and a path code. The Object Configuration Manager mappings for an environment provide answers to the following questions:

- Where are my logic objects processed?
- Where do my data objects reside?

The path code associated with an environment provides an answer to another question: Where do my Central objects reside?

OneWorld has the following types of environments:

- **Distributed data and distributed logic environments:** Determine where data resides and where application processing occurs.
- **Mode of processing environments:** Are designed to support the three types of transaction processing available with OneWorld: store and forward, direct connect, and batch of one.
- **Group of object environments:** Control which set of objects are used, such as production, development, or pristine.

4.2.1 Environment tables

A OneWorld environment definition involves the following tables, which are explained in more detail later in this section:

- Library List Control (F0093)
- Library List Master (F0094)
- Environment Detail—OneWorld (F00941)
- Object Configuration Manager (F986101)
- Path Code Master (F00942)

4.2.1.1 The Library List Control table (F0093)

The Library List Control table contains valid environments for each user. You must assign to each user at least one valid environment for signing onto OneWorld. The environments you assign to each user in the Library List Control table are validated at OneWorld startup against the OneWorld directories on the workstation. Only those environments that a user is authorized to sign onto and that are installed on the machine they sign onto are listed as available environments.

The Environment Revisions form of the User Profile application enables you to specify the users who are authorized to sign onto specific environments.

Note

Do not confuse authorization to sign onto an environment with authorization to install a specific package, which is handled through the Deployment Preferences form of the User Profile application.

4.2.1.2 The Library List Master table (F0094)

The Library List Master table contains the name of the environment and the description.

4.2.1.3 The environment detail: OneWorld table (F00941)

This table contains the environment name and the associated path code from the Path Code Master (F00942) table.

Every environment must have an associated path code. Environments can share the same path code. For example, two environments can use production objects and have different data location mappings. This is the case if you have a group of users processing against data on a corporate server and a group of users processing against data on a departmental server.

4.2.1.4 The Object Configuration Master table (F986101)

The Object Configuration Master table has data and logic object mappings for every environment. Every environment must have entries in the Object Configuration Manager table. This is true even if the mappings for the environments are identical. Two environments may have different path codes but have the same mappings for data and logic. For example, developers and testers can have different sets of objects, but their data is in the same database and their logic processes locally.

4.2.1.5 The Object Path Master table (F00942)

The Object Path Master table stores the location of the path code's Central objects, the release associated with this path code, and other details.

Figure 15 shows the tables used for defining a OneWorld environment and the relationships between them.

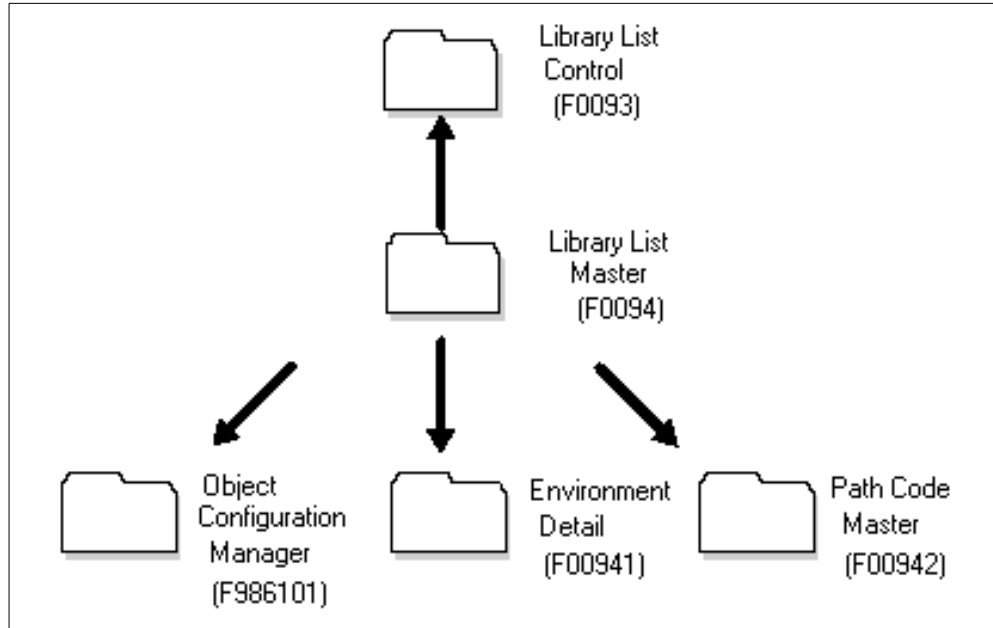


Figure 15. OneWorld environment definition

4.3 Path codes

A path code is a pointer to a set of objects. For each set of objects in your configuration, you must define a path code in the Object Path Master table. For

example, J.D. Edwards recommends a separate path code definition for each of the following sets of objects:

- J.D. Edwards pristine objects
- Production objects
- Development objects
- Conference Room Pilot (CRP) objects

The Object Path Master (P980042) application uses the Object Path Master (F00942) table.

4.3.1 At installation

OneWorld requires that you define a path code in the Object Path Master for each set of Central objects. A set of OneWorld objects consists of a central-objects data source and a directory of objects, which include business function source and include files, object files, and dynamic link libraries (DLLs). A path code definition contains the data source name of the central-object specifications and the directory path to the objects.

When you build a package for the workstation, you must specify a path code. OneWorld uses this path code to determine which set of Central objects to use as the source for the package and the directory to use as the destination for the package.

When you deploy objects to an Enterprise Server or to a workstation, OneWorld uses that path code to determine which set of replicated objects to push to the logic server. J.D. Edwards refers to objects stored on Enterprise Servers and workstations as sets of replicated objects. OneWorld stores replicated objects in directories and libraries using the same name as the path code from which they were deployed. OneWorld executes only replicated objects at runtime. If OneWorld opens an application that does not reside on the workstation and Just-In-Time Installation (JITI) is set for the workstation, the deployment data source installs the needed objects to the workstation at runtime.

4.3.2 At runtime

OneWorld uses path codes at runtime in the following ways:

- **To validate available environments:** When you sign onto OneWorld, it checks the path codes you defined in your environments against the path code directories that are physically installed on the workstation to which you are signed on. If that workstation does not have a path code that you defined in one of your environments, that environment is not displayed when you sign on to OneWorld.
- **To determine the directory location of a requested object:** OneWorld determines the directory on your workstation or server for the replicated objects. This is based on the path code of the environment chosen when the user signs onto OneWorld.

4.3.3 At development

When you check out an object for development, you use the Object Librarian to specify a path code. OneWorld uses the path code to determine where the central objects are stored and checks out the object from those locations (both database

and file server). The path code that you choose also determines the directory on your workstation into which OneWorld stores the replicated objects.

When you check in an object, you use the Object Librarian to specify a path code. OneWorld uses the path code to determine the workstation directory from which to pull the replicated objects and the location of the central objects in which to place the objects.

4.3.4 Locations

A path code can refer to the central development objects on the Deployment Server or to replicated objects on a workstation or logic server. A path code exists for each unique set of central objects. For example, you may have a set of objects reserved for software updates that you can deploy to users and a set of objects that you reserve for major enhancements.

A set of objects or the path code can reside in the following locations:

- **Central server:** Contains the central set of development objects specifications. This is the location where all development occurs. The path code connects the specifications and the C components on the Deployment Server.
- **Workstation:** Contains a replicated set of objects that OneWorld uses at run time.
- **Logic Server (or Application Server):** Contains a replicated set of objects that OneWorld uses to process logic on these servers.

4.4 Object Configuration Manager (OCM)

Object Configuration Manager (OCM) is a tool that configures distributed processing and distributed data at runtime without requiring programming. The Object Configuration Manager is the control center for your runtime architecture. As shown in Figure 16, OneWorld always uses the Object Configuration Manager to locate the data and the platform needed to execute the distributed logic.

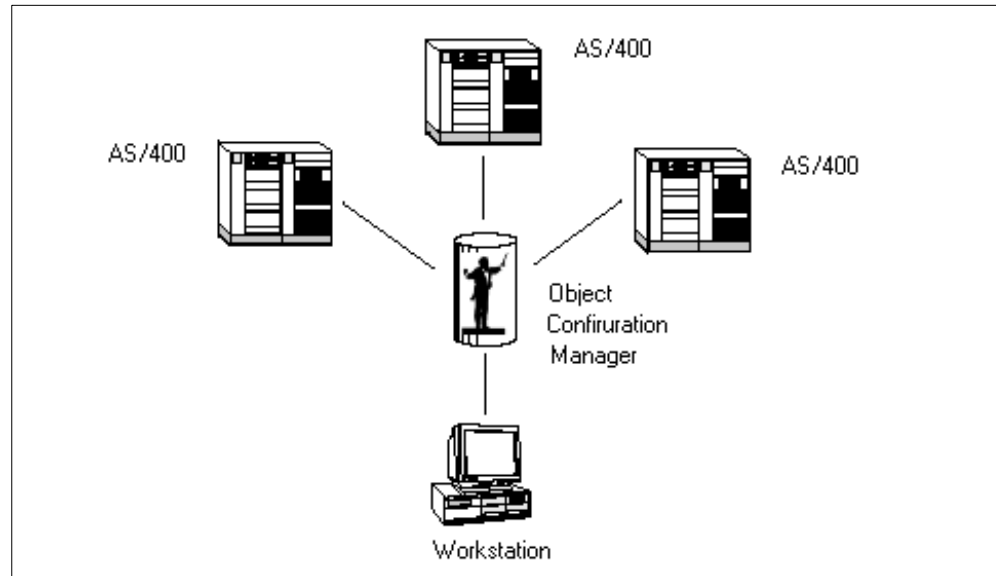


Figure 16. Object Configuration Manager (OCM)

4.4.1 Location types

The Object Configuration Manager (P986110) tool configures distributed processing and data dynamically without any programming. Depending on the environment and the user, the Object Configuration Manager points to the correct location for:

- Data
- Batch processes
- Business functions

The Object Configuration Manager stores information in tables that tell OneWorld where data resides and where processing occurs. At runtime, OneWorld looks to the Object Configuration Manager to determine these data and processing locations.

It may be helpful to think of the Object Configuration Manager as a policeman directing traffic or an orchestra conductor who directs several members to achieve a common goal. In OneWorld, business data objects (tables) map to database data sources. Batch processes and business functions map to machine data sources.

You always need at least two Object Configuration Manager tables:

- **One table for all workstations:** Normally, a central data server stores your system data source. If the central server is unavailable, OneWorld looks to the workstation's jde.ini file for a secondary location.
- **One table for each logic server:** Servers process differently than workstations. For example, the server map data source for each logic server stores separate Object Configuration Manager tables for server processing. The advantage of having this OCM is that it eliminates looking on another server to find that information is already on the server that initiates the request.

4.4.2 Retrieving application data

This section describes the steps for retrieving application data:

1. Any time an application requests data, OneWorld looks to the Object Configuration Manager table (F986101) for the data source of that table. For example, on the Work With Purchase Orders form, a user enters search criteria in the Query-by-Example line and clicks on Find. Before retrieving the information, OneWorld first has to locate the table in which the information (record) resides. To do that, OneWorld uses the Object Configuration Manager table to determine the correct data source.

The primary, unique index to the Object Configuration Manager table includes:

- Environment, such as PRD733 or DEV733
- User, which is either a specific user ID or group ID, or *PUBLIC
- Object name, such as F0101, B401002, or R09801
- Database path

Table 2 shows the sequential search hierarchy that the Object Configuration Manager uses to locate the correct data source for a data request. For this example, note the following points:

- The environment is PRD733 (production).
- The status is AV (active).
- The object type is TBLE (table).

Table 2. Example of data retrieval

Search sequence	Object name	User/group	Search criteria
1	F0101	SI5745669	Is there a record for the named environment, status active, type TBLE for the named table, and the specific user?
2	F0101	OWTOOL (Group ID)	Is there a record for the named environment, status active, type TBLE for the named table, and the specific group?
3	F0101	*PUBLIC	Is there a record for the named environment, status active, type TBLE for the named table, and *PUBLIC?
4	DEFAULT	SI5745669 (User ID)	Is there a record for the named environment, status active, type TBLE with no named table (default), and the specific user?
5	DEFAULT	OWTOOL (Group ID)	Is there a record for the named environment, status active, type TBLE with no named table (default), and the specific group?
6	DEFAULT	*PUBLIC	Is there a record for the named environment, status active, type TBLE with no named table (default), and *PUBLIC?

2. Once the data source is determined, OneWorld passes the definition of that data source (that is, database, platform, machine name, connect string, and library) to JDEBase.
3. JDEBase generates the SQL statement that is appropriate for the database being called. It works with third-party communications software to fetch the data and return it to OneWorld.

4.5 Data sources

OneWorld data sources are the building blocks that you use to set up a OneWorld enterprise configuration. They define where your database tables reside and where OneWorld runs logic objects for your enterprise. Data sources can point to:

- A database in a specific location (for example, a Microsoft Access database, such as JDEB7.MDB located in \B7\DATA, or an AS/400 data library, such as PRODDATA)
- A specific machine in the enterprise that processes logic

4.5.1 Defining data sources

Data source definitions are stored in the Data Source Master (F98611) table. Workstations use a common Data Source Master table, which generally resides in the system data source on the Enterprise Server. OneWorld servers that process logic and request data require their own unique definitions for OneWorld data sources. Therefore, they have their own Data Source Master in the server map data source.

There are at least two sets of the Data Source Master tables. They reside in a centralized system data source normally kept on an Enterprise Server and accessed by workstations, and in a server map data source, which each logic server requires.

4.5.2 Database data sources

A database is a grouping of tables in a database management system. You must identify databases to applications that access them. A database data source is a description of data location.

You can distribute databases across a network and involve various servers and database management systems. You must identify the database information (a database data source) to OneWorld to enable OneWorld to identify and connect to a database.

The following identifiers are important when defining a database data source on the AS/400 system:

- **Data source names:** The name of the data source you are defining. You should use a meaningful name for your data sources. For example, to indicate that you are storing business data for production users, your data source name can be "Business Data - Prod" or something similar.

J.D. Edwards provides default data source names at installation, which you can use for your own data sources. See the *OneWorld Installation Guide* for more information.

- **Library:** The library on the AS/400 system where the file or group of files resides.
- **Library list:** The relational database (RDB) name of the DB2 UDB for AS/400 database (WRKRDBDIRE).
- **DLL name:** J.D. Edwards buffers the business solution from the technology. The JDEBase middleware product provides a set of APIs that resolve SQL differences across multi-vendor database management systems. You must associate the proper JDEBase program with the data source definition. For example, you must use the JDBODBC.DLL with Microsoft SQL Server, DB2 UDB for AS/400, and Microsoft Access. These programs translate the generic Data Manipulation Language (DML) requests into appropriate SQL statements and work with the third-party communication software to communicate requests across the network.
- **Database name:** The name of the ODBC Data Source to be used in conjunction with this data source. The name must match the ODBC Data Source name exactly.
- **Server name:** The network node name of the machine where the database resides.

4.5.3 Logic data sources

A logic machine (also known as a *Logic Server* or *Application Server*) is the machine on which batch applications and master business functions are executed. A logic data source is a description of where OneWorld logic will be executed. You must identify these logic machines to OneWorld using a data source definition. The data source definition must include the network information about the machine, such as a server name like AS400A.

When mapping logic objects for distributed processing, it is the machine data source (distributed processing data source) that you use as the target location for processing logic objects.

4.5.4 Required logic and database data sources

There is a minimum number of data sources that you must set up for OneWorld to run. Two of the required data sources are logic data sources, while other data sources are database data sources.

The required logic data sources are:

- Distributed Processing
- Local

The required database data sources are:

- Business data
- Control tables
- Replicated local data
- Data dictionary by release
- System
- Object Librarian by release
- Central object specifications by release
- Server map

- Versions
- AS/400 data sources

Each of these data sources are described in the following sections. OneWorld installation software provides sample data sources to build your system configuration. See the *OneWorld Installation Guide* for details.

4.5.4.1 Required logic data sources

Two required data sources are:

- **Distributed processing:** This data source definition contains information that OneWorld uses to identify the logic machine in the network. You need to define each logic machine as a data source.
- **Local:** This data source defines the OneWorld workstation. Use this data source to override the process location of a batch application that you mapped in the Object Configuration Manager to run on the server.

4.5.4.2 Required database data sources

This section describes the required database data sources:

- **Business data:** This data source is used to point to the actual location of your business data. The business data can be divided into multiple owners or libraries, which can reside on the same Enterprise Server or on different ones. Each group of data requires a separate data source.

Some examples of business data include:

- Production data (non-technical data, such as financial and manufacturing data)
- Test data
- Demo data (demonstration or training data)
- Conference Room Pilot (CRP) data

The OneWorld installation procedure provides demonstration data that you can copy to supported host databases. The data source name is Business Data - JDE.

- **Control tables:** This data source consists of user-defined codes, menus, and next numbers.
- **Replicated local data:** This data source points to the Microsoft Access database you use for edits made on the workstation. The amount of replicated local data depends on the transaction model you choose (for example, store and forward rather than direct connect). The user-defined codes and menu tables should reside in a Microsoft Access database file on the workstation. Replicated local data is by path code and stored on the workstation in the b7/path code/data path.
- **Data dictionary by release:** This data source allows you to store OneWorld Data Dictionary master tables in a central location to allow easier administration of changes. Group these master tables together to form a Data Dictionary database. You should share one Data Dictionary between your production (such as PRODB733) and development (such as DEVB733) path codes. OneWorld allows one Data Dictionary per path code, but multiple data dictionaries are not recommended or supported.

- **System:** This data source consists of the technical tables you use to run all OneWorld applications. You need to set up one system data source per release. When running applications, the system tables provide the following information:
 - Environment detail (map these to the AS/400 system for coexistence with WorldSoftware)
 - User profiles and preferences (map these to the AS/400 system for coexistence with WorldSoftware)
 - Object mappings (location of tables, batch processes, and business functions)
 - Data source definitions
 - OneWorld security

All workstations use a central set of system tables usually stored on your Enterprise Server, but not on your Deployment Server. Each logic server requires its own subset of system tables. These server system tables are stored in the server map data source. See the “Server Map” bullet.

When a user first signs on, OneWorld uses the user ID and environment to retrieve information from the system tables for that user and environment. OneWorld caches this information in memory on the workstation. Anytime a change is made to the central system tables, dynamic caching of the system information occurs for those workstations with an active OneWorld session.

When OneWorld starts up on a workstation, OneWorld attempts to connect to the base data source found in the workstation jde.ini file. If this data source is unavailable, OneWorld attempts to connect to a secondary data source for system information if it is defined. If the secondary data source is implemented, it is important to have processes for ensuring that the alternate system data source location contains current information. You can maintain an alternate data source's information using Table Conversion or data replication.

The jde.ini file should look like the following example for the primary system data source connection:

```
[DB SYSTEM SETTINGS]
.
.
Default Env=DEMOB7A
Default PathCode=DEMO
Base Datasource=System - B733
Database=System - B733
.
.
```

The Secondary System Data Source connection is shown here:

```
[DB SYSTEM SETTINGS - SECONDARY]
Base Datasource=Access32
Object Owner=
Server=
Database=Access32
Load Library=JDBODBC.DLL
```

During installation, the system data source is tied to a release with the Release Master application. Configuring the release updates the setup.inf file used during the workstation installation to create the jde.ini file.

- **Object Librarian by release:** This data source points to the Object Librarian tables you use for custom development. You should have only one set of Object Librarian tables per software release, regardless of how many path codes (sets of Central objects) you maintain. This data source can reside on any J.D. Edwards-supported platform.
- **Central object specifications by release:** This data source points to the OneWorld source objects (central object specifications), as well as the User Overrides (F98950) table. Central-object data sources are databases.

If you have multiple path codes, they each need a separate central-objects data source. Developers check out objects from a central-object data source for modification. When the developer checks in the objects, the system copies the objects from the developer's workstation to the relational database tables in the central-object-specification data source. You must set up one central objects data source for every path code needed in your configuration. For example, you must have a central-objects data source for:

- Pristine objects
- Production objects
- Development objects

You connect each central objects data source to a path code used by the environments that you created for your configuration. This data source may reside in DB2 UDB for AS/400, an SQL server, or an Oracle database, and can reside on any J.D. Edwards-supported platform.

Regardless of which database is storing the relational database (RDB) components of the Central objects, you must store the C-language components on the Deployment Server that is accessible to all workstations for deployment and redeployment.

- **Server map:** This data source enables you to create for each logic server its own subset of system tables, called *server map tables*. Server map tables are required for each logic server. You must maintain these tables to ensure the integrity with the workstation's system tables.

Use server map data sources to establish unique object mappings for logic servers. When batch jobs and business functions running on the server request data, they look to the Object Configuration Manager and the Data Source Master tables in the server map data source. This is necessary because the mappings are different.

For example, suppose a user signs on to an environment that maps static local data on the workstation, dynamic transaction data to the server, and the master business functions and batch processes to the server. The user enters a sales order. The user edits user-defined codes, tax rules, and other static data against the workstation's replicated data. The user clicks OK to enter the order, which executes the Sales Order Entry master business function on the server. The master business function also needs transaction data and some of the replicated data. It does not make sense for the master business function to go back to the workstation to retrieve user defined codes and tax information. Therefore, the server map Object Configuration Manager table maps all data to the appropriate server data source. More than likely, the server user-defined codes, tax rules, and so on are the published tables from which the workstation's tables are replicated. See "Data Replication" in the *OneWorld System Administration Guide* for information about replicating tables.

The following tables in the Server Map database are unique to a server's perspective of processing:

- **Data Source Master (F98611)**: To enable servers to request data from other servers, you must assign the proper JDEBase program to the data source definition in the Data Source Master file in the server map data source.

For example, when a workstation requests data from an Oracle database, the data source definition for the Oracle database uses JDBOCI32.DLL. The third-party product called Oracle Call Level Interface (OCI) performs the communications between workstation and server. When an AS/400 system requests data from an Oracle database, the data source definition for that database should use JDBNet, which manages communications.

- **Object Configuration Manager (F986101)**: Logic objects processing on a server request data and perhaps other logic objects. When these requests are made to OneWorld running on a server, Object Configuration Manager must be accessed to find the correct mappings for the data and logic objects. Servers may have different mapping requirements than workstations.

For example, you should map all user-defined codes locally to the workstation for performance during interactive processing. Server processing may require you to map these files locally to a server database to enhance performance of server processing.

- **Job Control Status Master (F986110)**: Records information about batch jobs launched on a server.
- **Job Number Master (F986111)**: This is the next number table for batch jobs launched on a server.
- **Data Replication Change Log (F98DRLOG)**: Stores one record per data change and stores the actual changed data. Each server that has publisher tables must have this table in its server map data source.
- **Data Replication Pending Change Notifications (F98DRPCN)**: Stores one record per data change per subscriber. This log records unprocessed change deliveries so that OneWorld can know which subscribers have not yet received the change deliveries. Each server that has publisher tables must have this table in its server map data source.

- **Versions**: This data source corresponds to the path code, as in Versions - PRODB733. It stores versions and processing option information. It includes the following tables:

- Versions List (F983051)
- Processing Option Text (F98306)

- **AS/400 data sources**: Additional data sources are required for users who have an AS/400 Enterprise Server. You must identify these data sources as Do Not Translate (DNT) because they contain OneWorld tables with binary large objects (BLOBs). BLOBs are used by OneWorld for security and performance purposes. They cannot be translated into a form directly readable by a database other than OneWorld.

The data source master definition for DNT data sources must have the AS/400 BLOB Data Source field checked. These are the data sources:

- **System - DNT:** This data source stores the following non-translated tables:
 - OneWorld Security (F98OWSEC) table for sign on security
 - Data Replication Change Log (F98DRLOG)
 - Data Replication Pending Change Notifications (F98DRPCN) log
- **Versions - DNT:** Stores, by path code, the Versions List (F983051) table.
- **Business Data - DNT:** This data source corresponds to the environment, and stores the Workflow Activity Specifications (F98811) table.

4.6 Object storage

OneWorld provides two general object storage formats, *source objects* and *runtime objects*, to accommodate several functions in OneWorld.

Source objects (often referred to as Central objects) consist of object specifications for each OneWorld object and C component. They are stored in a central location to allow deployment, redeployment, and development. Central objects are located in a relational database on either an Enterprise Server or a Deployment Server, depending on available resources.

Runtime objects consist of local specifications (Specs), as well as linked and compiled C components. Runtime objects are stored on Logic Servers (also known as Application Servers) and are used for day-to-day processing.

To deploy objects to the Enterprise Server, you define a package that OneWorld creates from source objects. Each package contains a copy of the runtime objects. OneWorld converts the source object specifications to a format for storage in a directory (Specs or runtime objects). Workstations and enterprise logic servers receive packages that contain runtime objects (among others) and store them in local directories. Figure 17 shows how source objects are replicated to Specs (runtime objects) on locations that run OneWorld logic.

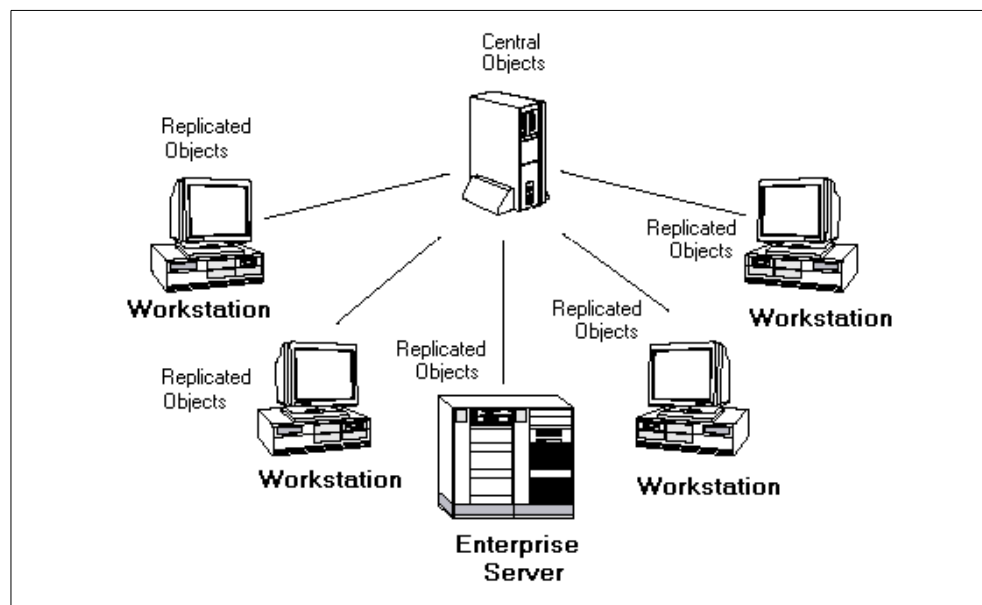


Figure 17. Central objects replication

At runtime, OneWorld requires the specifications of the object that the workstation or enterprise logic server processes.

On a OneWorld workstation or enterprise logic server, you can store one set of runtime objects for each set of source objects. For example, your enterprise may use separate sets of central objects for a development environment and a production environment. A separate development environment provides easy distribution of custom modifications and maintains the integrity of objects in use by other environments within your enterprise.

4.7 Object deployment

Deploy OneWorld to your workstations and servers using any of the following methods:

- Initial installation for workstations and servers
- Workstation installation for workstations
- Application installation for workstations
- Just-In-Time Installation for workstations
- Server package installation

4.7.1 Initial installation

The installation process is based on a centralized Deployment Server model. The Deployment Server installation program copies OneWorld installation software from the CD-ROM to the Deployment Server. From the Deployment Server, you redistribute the software to the Enterprise Servers and workstations.

4.7.2 Workstation installation

The Workstation Installation program retrieves software from the package you request. A package contains instructions that describe where to find the necessary components that the Workstation Installation program deploys to the local computer.

Each package represents a record of your central objects at a point in time. Once you build and test a package, you can safely modify central objects, because users will not receive those objects until you build another package and make it available to them. Building a package involves copying the central objects to the package itself. The package then contains replicated objects, which OneWorld can read at runtime.

4.7.3 Application installation (update packages)

Application installation can be used to quickly deploy changes to an individual application. The advantages of application installation are:

- Users do not have to install a complete set of objects, but only the objects necessary to capture the changed application.
- You do not need to build a new package and perform a global build before deploying the application change.
- Developers and testers can use application installation to load changes onto their machine that were recently checked into the central objects.

4.7.4 JITI

Just-In-Time Installation (JITI) provides applications that do not currently reside on your machine when the situation arises that you need them. For example, when you deploy a custom menu that contains a new application to a workstation, the object automatically installs on the workstation when a user clicks the menu option for the application.

4.7.5 Server packages

All OneWorld application development takes place on a workstation with objects stored on a Deployment Server through Object Librarian. OneWorld allows you to put business applications to an Application Server. Server Package Installation enables developers and system administrators to push objects, called a server package, from the central objects location to Enterprise Servers.

A server package is a group of OneWorld objects that you have grouped together with a common Software Action Request (SAR) number. Objects consist of specification records, source files, and header files. Compiled objects are created on the Enterprise Servers. After you define a package, you can install it to multiple servers and multiple path codes.

You can install the following objects on an Application Server:

- Business functions
- Business views
- Data structures
- Tables (installation does not create the table in a database; it only pushes the specifications and table header files to the server)
- Batch application specifications (both templates and versions)
- Application specification records

Server packages are not the same as the workstation package built through Package Revisions. However, there is a processing option for Package Revisions that allows you to automatically create a corresponding server package when you create an update package for workstations. J.D. Edwards recommends that you use this feature to ensure that you are deploying the same objects to your Enterprise Server that you deploy to workstations.

4.8 Middleware

In a client/server environment, applications must communicate across different platforms. These platforms can have different communications protocols, database management systems, and hardware operating systems. For clients to talk to servers and servers to talk to other servers, a mechanism must exist that can bridge multi-protocol and multi-vendor issues. This mechanism is a layer of software called *middleware*, which resides between the operating system and the business applications. It is important to have an application architecture that is based on a single, consistent middleware strategy.

J.D. Edwards provides the following types of middleware:

- **JDENet Communication Middleware:** JDENet Communication Middleware performs the connections from workstation to server and server to server, and

sends messages for distributed requests. It is a peer-to-peer, message-based, socket-based, and multiprocess communication middleware solution.

- **JDEBase Database Middleware:** JDEBase Database Middleware provides platform-independent application program interfaces (APIs) for multidatabase access. These APIs are used in two ways:
 - By OneWorld applications that dynamically generate platform-specific Structured Query Language (SQL), depending on the data source request.
 - As open APIs for writing advanced business functions in the C language. OneWorld uses these APIs to dynamically generate platform-specific SQL statements.

JDEBase also provides workstation-to-server and server-to-server database access. To accomplish this, OneWorld is integrated with a variety of third-party database drivers, such as the IBM Client Access/400 database software, and the Microsoft Open Database Connectivity (ODBC) programming interface.

4.8.1 JDENet

JDENet is the J.D. Edwards proprietary communications middleware package that provides for server-to-server and workstation-to-server communication. The features of JDENet include:

- Socket_based communication
- Message_based communication
- OneWorld process_based design

JDENet is a peer-to-peer middleware solution. Think of a client as a network conversation initiator and a server as a network conversation responder. A client always initiates the conversation by asking for something from another machine. A machine acts as a server when it responds to a network request, such as when it gives something asked for by the client. In this peer-to-peer middleware solution, the distinction between client and server is determined by which machine starts the conversation. Any machine, running on any platform, can act as a client or a server at a given time.

With JDENet, communication between client and server occurs through messages. JDENet messages contain requests for processing such as requests for business functions, batch jobs, data replication, or OneWorld signon security. JDENet messages can originate from the client or the server. JDENet handles database requests only if multiple servers are in use and if they are different server types.

Application requests (messages) can be synchronous or asynchronous. A synchronous message, such as calling a business function, requires the client to wait for the server to complete the request. An asynchronous request, such as a batch process, enables the client to continue with another task, while OneWorld processes the request. In some circumstances, business functions can also be called asynchronously.

4.8.1.1 Socket-based communication

Sockets provide a duplex communication channel between one or more systems. JDENet uses stream sockets to provide end-to-end communications. Sockets guarantee that the data arrives intact.

4.8.1.2 Message-based communication

Message-based communication means applications send service requests for logic or data in the form of messages that are received and stored in a queue for processing. The middleware handles message transmission, which enables the workstation application to process other tasks. Without messaging services to handle these jobs, the application must wait until the request is handled and the results returned.

Messaging is most appropriate for applications that are event driven. It is the opposite of remote procedure calls (RPC), which are synchronous. The message packaging and handshaking of JDENet ensures that the message transmission is complete.

4.8.1.3 OneWorld process-based design

Although OneWorld workstations can have more than one copy of OneWorld loaded, only one OneWorld Windows-executable application can run at any one time. OneWorld uses an internal network process (also referred to as a *net process*), called JDENet, to communicate a request to the OneWorld server.

Servers also have a net process called JDENet. This process communicates with the workstations and routes request messages to the appropriate dedicated OneWorld processes. In turn, the dedicated processes route work to the appropriate platform-specific logic processes, such as DLLs, shared libraries, and job queues. A server can have multiple OneWorld main processes, multiple OneWorld dedicated processes, multiple DLLs, shared libraries, and job queues.

The advantage of this architecture is that multiple workstations can make requests to the same server at one time. You can control the number of workstations that can make and maintain a session connection to a main server process. You can also define the total number of dedicated processes (and the number of each type) that OneWorld uses to process specific types of workstation requests.

Network processes

There is a relationship between the net processes, the dedicated processes, and the logic processes. This relationship is specifically defined by the `jde.ini` file on the Enterprise Server. Every Enterprise Server must have at least one OneWorld network process, referred to as a JDENET_n job. This job handles network connections and traffic for OneWorld.

As defined in the `jde.ini` file for each server, there can be multiple JDENET_n processes. Regardless of the number of JDENET_n processes, a single (the initial) JDENET_n process serves as the “master listener”.

If multiple JDENET_n jobs are specified, OneWorld starts the jobs as required, allocating a job to each request. When the maximum number of JDENET_n processes are started, OneWorld automatically alternates between the currently running JDENET_n jobs until the maximum number of connections is reached. This provides a degree of load balancing between OneWorld network processes. If, on a given server, the maximum number of connections for the JDENET_n job is met, a client or server cannot initiate an additional OneWorld session on that server until an existing session connection is ended. This is because, by design, all connections to JDENET_n persist for the duration of a OneWorld session.

For example, assume the `jde.ini` file on the server specifies that four `JDENET_n` processes are allowed. The first `JDENET_n` request is routed to the “master listener”, which is the initial `JDENET_n` process that is run at server start-up. When a second request to `JDENET_n` is received, the “master listener” receives the request and assigns it to a second `JDENET_n` process, which it then starts. This assignment persists for the duration of the session between the requesting device and this server. The same process occurs for the third and fourth `JDENET_n` requests. When the fifth request is received, that request is assigned to the first `JDENET_n` process, and the cycle continues.

Kernel processes

The responsibility of the `JDENET_n` process is to handle the network layer of communication. If the `JDENET_n` job determines that the incoming message is a request for logic processing, it routes the request to an appropriate `JDENET_k` job. OneWorld determines an appropriate `JDENET_k` job based on message identifiers. The `JDENET_k` job is the OneWorld process that provides the link between the `JDENET_n` job and the appropriate platform-specific processing job. The `JDENET_k` process applies only to servers.

The `JDENET_k` job handles the two-way routing to and from the various logic processes. The `JDENET_n` job handles the return delivery to the appropriate machine. There are 11 dedicated kernel types, each of which is responsible for a specific type of OneWorld process.

Examples of logic processes include dynamic link libraries (.DLL) for Windows NT platforms, shared libraries (.sl) for UNIX platforms, and JDENet processes for AS/400 platforms.

For more information about JDENet middleware, see “Middleware” in the *OneWorld System Administration Guide*.

4.8.2 JDEBase

Different database management systems (DBMS) have their own version of SQL. A database middleware layer allows a common interface to interpret the various versions of SQL. J.D. Edwards has a database middleware product called JDEBase, which is a common set of application programming interfaces (APIs) that programmers can call to request data and perform data manipulation logic. JDEBase interprets the generic APIs and converts the SQL into the appropriate statements for OneWorld to access the database.

Multiple databases in a distributed environment require a monitoring program to ensure database integrity. This monitoring program is referred to as a transaction monitor. The JDEBase database middleware has an embedded transaction monitor.

JDEBase provides:

- Ability to insulate developers from platform-specific SQL coding
- Rapid development of native drivers
- Server-to-server communication
- Transaction processing

JDEBase provides a set of APIs to the developer and a set of translation programs to OneWorld. The translation programs are embedded in the data

source definitions. For example, suppose a data request for Address Master is made. The Object Configuration Manager (OCM) determines which data source contains the requested table. The data source master provides the database information. This includes the required driver, which is loaded to translate the request into the appropriate SQL statements.

For example, the user enters search criteria for a Purchase Order and clicks on Find. The following series of actions occurs:

1. OneWorld sends a data request to JDEBase on the workstation. The request includes the form's data structure and any values needed to locate the record.
2. JDEBase performs these actions:
 - a. Builds data structures from the application structures, creates the actual SQL statement or equivalent commands, and passes it to the third-party communication software, which resides on the workstation. Examples of third-party communication software are IBM Client Access/400, Microsoft Open Database Connectivity (ODBC), or Oracle SQLNET.
 - b. Manages the physical connections to the database.
 - c. Manages optimal fetch algorithms.
 - d. Performs all binding.
 - e. Passes result codes back to the application. Logs errors in the log files (jde.log).
3. The third-party communication software on the workstation passes data to the third-party communication software on the server. The server accesses the table and returns the data back to the third-party communication software.

Chapter 5. OneWorld implementation designs

Since OneWorld was designed to be flexible, it allows many different implementation design possibilities. Using examples, this chapter looks at how different implementation designs can be used in a variety of situations. For each of the designs, we provide a list of pros and cons to help you determine which design is most suitable for your particular customer situation. Generally, the more *centralized* the design is, the lower the manageability costs should be. The more *decentralized* the design is, the lower the hardware costs tend to be.

5.1 Terminology

This section explains the terminology used in J.D. Edwards OneWorld implementation designs.

5.1.1 Different types of servers

The following list describes different types of OneWorld servers. Keep in mind that these servers can be on separate machines or combined on a single system. The servers include:

- **Database Server:** Database Servers have been around for many years. They run the Database Management System (DBMS), hold the transactional data, and service data related requests.
- **Application Server (also known as Logic Server):** An Application Server is used to handle business logic requests for one or more end users. These servers offer part of the scalability of the design since more of them can be added as the number of users goes up. A large installation may have many Application Servers.
- **UBE (Universal Batch Engine) Server:** The UBE Server handles requests for reports and batch updates. In many designs, the UBE Server is on the same machine as the Database Server. This makes good sense since the data needed for the reports and batch updates is located there. If scalability becomes an issue, the UBE Server may be separated from the Database Server.
- **Enterprise Server:** A OneWorld Enterprise Server is a machine where the Database Server and Application Server functions reside. It may also act as the UBE server. Enterprise Server represents a more host-centric design approach.
- **Workgroup Server:** A Workgroup Server is a localized data server holding non-volatile data for a set group of users. This server is often in a remote location (separated by a WAN) from the Database Server. It is normally on the same LAN as the clients that it is serving. Its main function is to reduce WAN traffic and improve performance for remote users.
- **Deployment Server:** A Deployment Server is the location where OneWorld is initially installed. This is the location where the deployment plan is entered to describe the architecture to be used. It is also the location where clients attach to load client packages.

5.1.2 Distributed processing and distributed data

OneWorld offers the ability to distribute processing and data. This allows the implementation design to accommodate moving the data to where the processing is executed and moving the processing location to where the data is located. Performance tuning possibilities are greatly increased in a network-centric design.

5.1.2.1 Distributed processing

In a network-centric design, the advantage is having the ability to move different types of processing to a location where they can be executed most conveniently and efficiently. The various types of distributed processing include:

- **Presentation processing:** Presentation processing determines the “look and feel” of the user interface. It captures external events and may perform some degree of editing incoming data. In OneWorld, this is done through Windows APIs or a Web browser to achieve the GUI interface. This logic almost always runs on the client and has to run on a machine running Microsoft Windows.
- **Business processing:** Business processing represents the execution and enforcement of the business rules of the software application. It also executes business related calculations. Generally it executes for short periods of time before returning control to the application user for more keyboard interaction. In OneWorld, this processing is accomplished through business functions and can have its execution location mapped through the OCM.
- **Data management processing:** Data management processing provides access to corporate data. It handles database reads, writes, and data locking issues. Commitment control may also be handled here. In OneWorld, this function is handled by JDEBase middleware and the DBMS of the platform such as DB2 UDB for AS/400.
- **Batch and report processing:** Batch and report processing performs actions on large numbers of records within the database. It has a tendency to execute for extended periods of time without the need for intervention. It is also usually associated with large numbers of database I/Os.

5.1.2.2 Distributed data

In a client/server environment where locations are separated by WAN links, the ability to relocate data closer to the user becomes an important performance advantage. OneWorld's ability to map data by user through the OCM makes it easier to make these performance adjustments.

This process usually involves the replication of data to a Workgroup Server located near the users of that data. An analysis needs to be done for each application to determine which portions of the data are eligible for replication. Distributed data includes:

- **Control data:** Control data is non-volatile data that is accessed many times during the running of a specific application, but is rarely changed. This type of data is a good candidate for replication to a Workgroup Server. An example of this type of data is Master Chart of Accounts.
- **Transactional data:** Transactional data is volatile data that is changed almost every time it is accessed. This type of data is usually not a good candidate for replication and may be better left in a centralized location. An example of this type of data is General Ledger Detail.

5.2 Implementation design examples

This section shows examples of various implementation designs with pros and cons of each design. The examples shown are simply that—examples. An example does not represent the only possible implementation design in a given customer situation.

5.2.1 Heavy client

The heavy client (also known as the fat client) solution allows for full OneWorld functionality. The client contains a full set of objects and processes all of the logic locally on the client's workstation. Heavy client is a scalable solution with a relatively low startup cost.

The heavy client needs to run a Windows 95, Windows 98, or Windows NT workstation. Because of package deployment needs, manageability becomes more complex for a large number of users.

With this solution, the traffic between the client and the server is quite intensive, which can cause performance problems in a Wide Area Network (WAN) environment. The traffic can be reduced for intensive data-entry applications when implemented with Workgroup Servers (see 5.2.1.3, "N-tier design" on page 79). However, the traffic can not be reduced for transactional inquiries. In this case, data replication is usually needed, which often requires a database administrator and drives up the costs. There is less need for a database administrator on the AS/400 platform due to the integrated nature of the AS/400 database.

Mixing two-tier, three-tier, and n-tier heavy client designs in one environment allows the moving of data close to the logic execution location and the logic execution location close to the data.

5.2.1.1 Two-tier design

In a pure two-tier design (Figure 18 on page 74), the Database Server handles only data management processing, and the client handles all other (presentation, business, and batch processing) processing. In some cases, batch processing is moved to the Database Server so that batch processing and reports can be executed as an unattended process.

It is possible that a two-tier design may not perform well in a WAN environment with remote clients. This is because all data requests must move back and forth over the WAN link. In this case, network latency can become a serious issue.

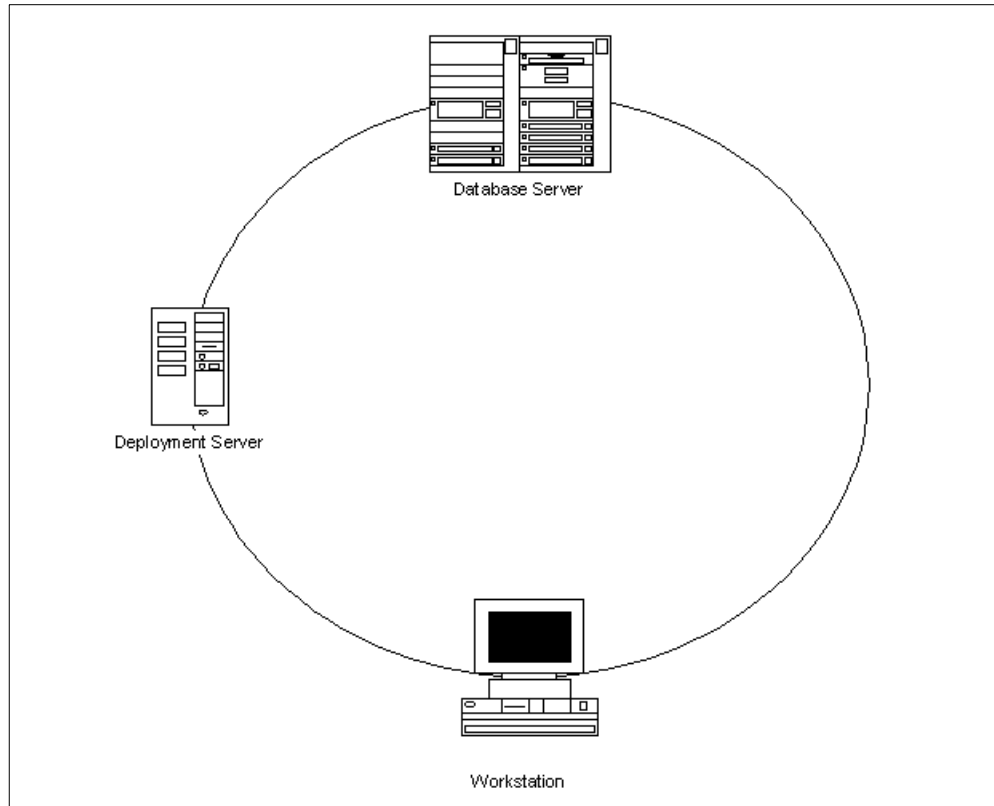


Figure 18. Two-tier design

Pros and cons of a two-tier design

The two-tier design has some advantages and disadvantages compared to other designs described in following sections:

• **Pros**

- Allows the most functionality of all the configurations including application development.
- Allows business logic, including UBEs, to be run on the client if desired.
- Relatively inexpensive since no Application or Workgroup Servers are required.

• **Cons**

- Requires a larger and more powerful client than the light client solutions.
- The performance of this configuration is generally not acceptable for use over the WAN.

Using the AS/400 system in a two-tier design

In a two-tier design, the AS/400 system acts as a Database Server. It can also act as the Deployment Server (Figure 19), while the remainder of the processing takes place on the client.

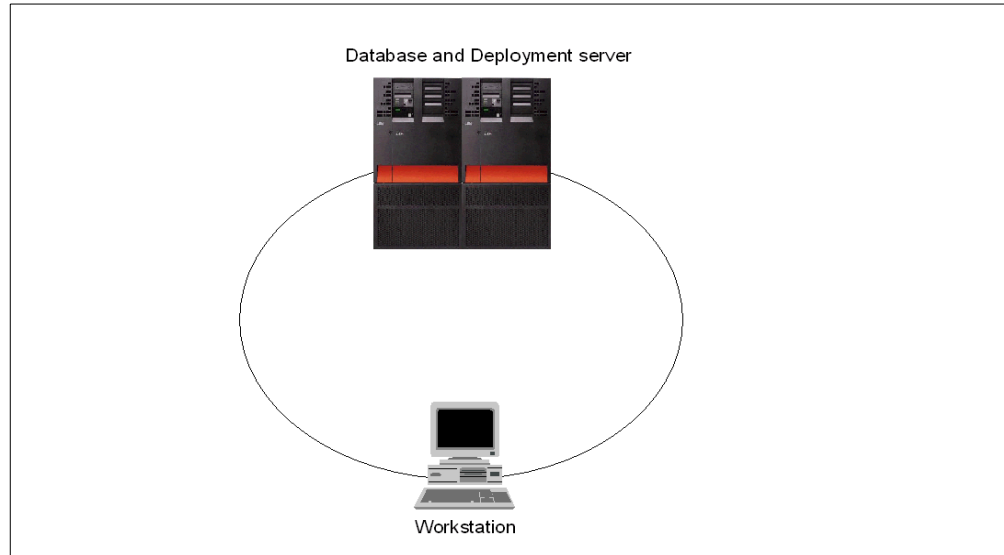


Figure 19. Two-tier design with the Deployment Server on the AS/400 system

The Deployment Server in Figure 19 runs on the AS/400 Integrated Netfinity Server, which is an integrated PC, residing in the AS/400 box. For more information on Integrated Netfinity Server, refer to Chapter 18, “The Integrated Netfinity Server and OneWorld” on page 513.

For more information on the pros and cons of using the Integrated Netfinity Server as a Deployment Server, refer to 18.3.3, “Integrated Netfinity Server as a Deployment Server” on page 519.

5.2.1.2 Three-tier design

A three-tier design is similar to a two-tier design, except that an Application Server is added to handle some of the processing that, in a two-tier design, is handled by the client. Normally, the client continues to perform all presentation processing, the Application Server handles business processing, and the Database Server continues to handle data management processing. Batch processing can be accomplished on either the Application Server or the Database Server.

The three-tier design (Figure 20 on page 76) should perform better over the WAN than the two-tier design. The reason is because the Application Server, which is located at the central site near the Database Server, handles most of the data access.

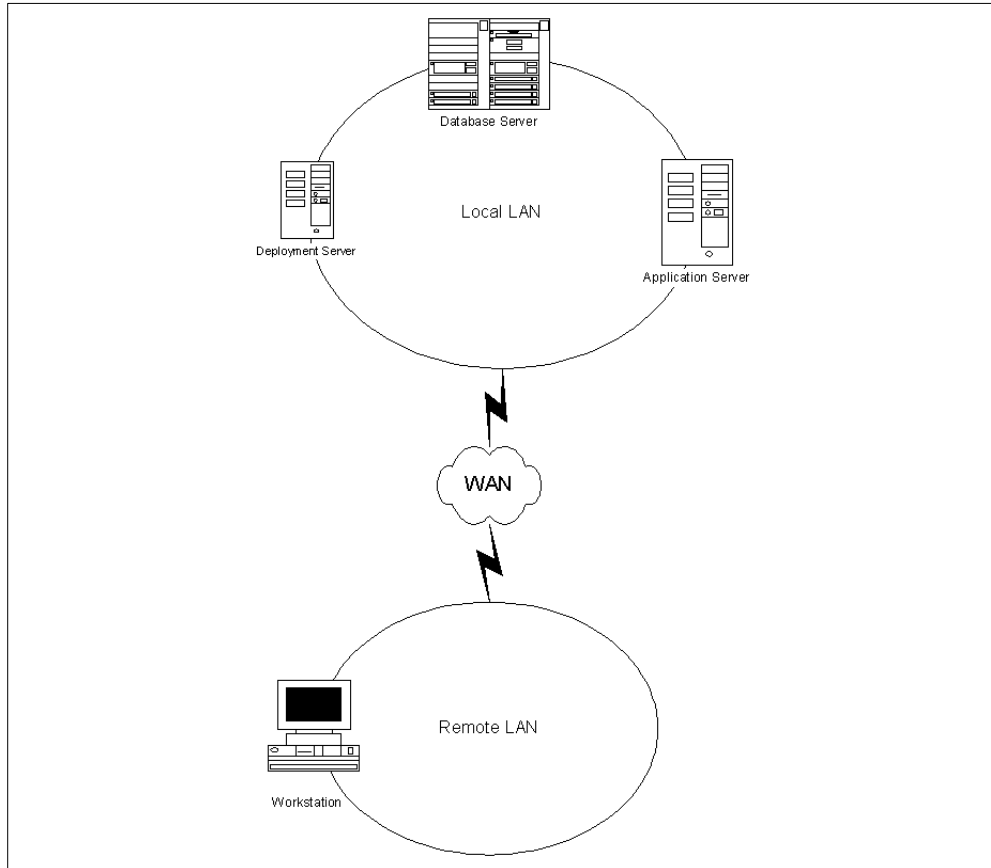


Figure 20. Three-tier design

Virtual three-tier design

With the larger Enterprise Servers, such as the high-end AS/400 systems, it may be desirable to have the Database Server and Application Server reside on the same machine (Figure 21). This is called a *virtual three-tier design*. This same machine may also perform batch processing.

Using the AS/400 system in a three-tier design

Due to its superb scalability, in many customer scenarios, the AS/400 system can perform well as an Enterprise Server, where one AS/400 system runs a Database Server, UBE Server, and Application Server as shown in Figure 21 (a *virtual three-tier design*).

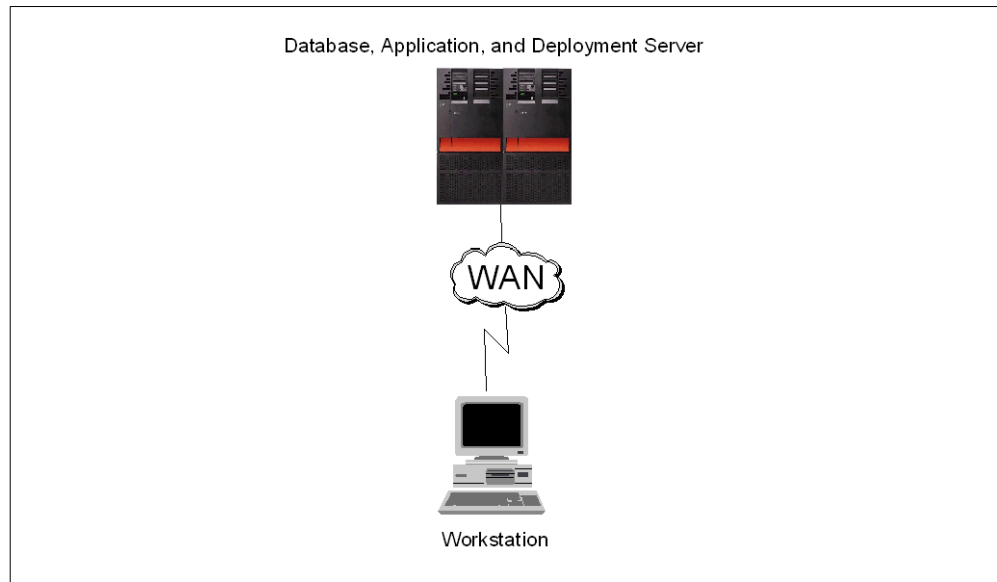


Figure 21. Using the AS/400 system in a three-tier design

In a *physical* three-tier design, one or more AS/400 systems can be configured as Application Servers communicating via OptiConnect or TCP/IP with the AS/400 Database Server. If required, UBE functions can be isolated to separate AS/400 servers as well.

With logical partitioning (refer to 2.9, “Logical Partitioning (LPAR)” on page 18, for an explanation of logical partitioning), OneWorld applications and functions can be assigned to individual AS/400 logical partitions to support the need for separate, dedicated systems within one multi-partition system unit. Reasons for using LPAR in OneWorld environment can include:

- Supporting multiple time zones on a single system
- Setting hard limits on the allocation of system resources. For example, a logical partition can be established for dedicated Universal Batch Engine processing. This would keep batch usage from encroaching on the CPU resources of other OneWorld processes.
- Dedicated OneWorld environments (testing, development, etc.) on a single system
- Supporting multiple character sets (Single Byte and Double Byte Character Sets) on a single system
- Increased availability: Backing up an environment to another partition using the command Save Restore Object may be faster than backing up to tape media
- Support for multiple releases of OS/400 or OneWorld on a single system

For more details on LPAR customer scenarios, refer to 2.9, “Logical Partitioning (LPAR)” on page 18. For additional information, requirements, and instructions on how to create logical partitions, refer to *Slicing the AS/400 With Logical Partitioning: A How To Guide*, SG24-5439.

Pros and cons of a three-tier design

The Three-tier design has some advantages and disadvantages compared to other designs:

- **Pros**

- Remote application development is possible. This allows programmers from different locations to work on the same development system.
- Performance is generally better than on a two-tier design over the WAN since most data validation functions can be performed on the Application Server that is located on the Local LAN.
- Lends well to a virtual three-tier design, which eliminates the need for a separate Application Server.

- **Cons**

- Is generally more expensive than the two-tier design since an Application Server must be purchased.
- The performance over a WAN is still probably not acceptable since most data traffic must travel over the WAN link.
- Client packages must be delivered over the WAN, which makes package management more complex. A remote file server may be needed to make the package available to all workstations on the remote LAN.

Pros and cons of a virtual three-tier over a physical three-tier design

The possible pros and cons of a *virtual* three-tier over a *physical* three-tier configuration are:

- **Pros**

- Cost savings due to less physical hardware: Software licensing, hardware acquisition and maintenance costs, and electrical power usage.
- Less floor space and heating and cooling requirements.
- Ease of implementation: There are fewer physical components to configure and integrate.

- **Cons**

System resource capacity limitations: You are limited by that system's maximum configuration specifications.

Pros and cons of a physical three-tier over a virtual three-tier design

The possible pros and cons of a *physical* three-tier compared to a *virtual* three-tier configuration are:

- **Pros**

- Increased system resource capacity
- Flexibility in locating work where system resources are available

- **Cons**

- Cost increase due to more physical hardware: Software licensing, hardware acquisition and maintenance costs, and electrical power usage.
- More floor space and increased heating, cooling, and power requirements.
- Higher complexity of implementation: There are more physical components to configure and integrate.

5.2.1.3 N-tier design

The n-tier design (Figure 22) adds Application Servers at the central location, as well as Workgroup Servers in the remote locations near the remote users. This design allows the most flexibility since data can be moved close to the processing, and processing can be moved close to the data.

Although this design is more complex, it should give the best performance of the heavy client design in a WAN environment since data and processing can be located where they are needed. A data replication product is required to keep the Master Database (the database on the Database Server) and Subscriber Database (the database on the Workgroup Server) in synchronization.

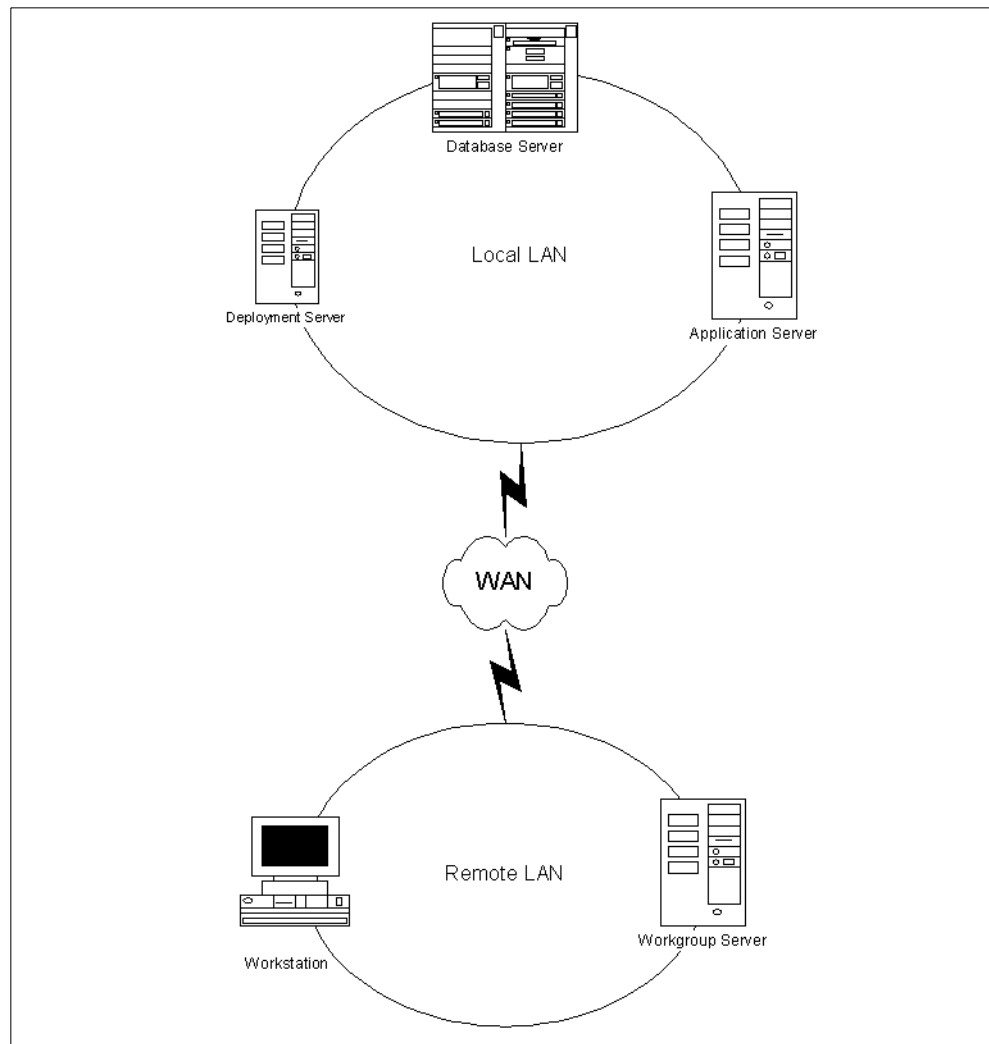


Figure 22. N-tier design

Pros and cons of an n-tier design

The advantages and disadvantages of an n-tier design are:

- **Pros**

- Remote application development is possible. This allows programmers from different locations to work on the same development system.

- Performance is generally better than three-tier over the WAN since static data files can be stored on the Workgroup Server on the remote LAN and accessed by all workstations located there.
- Lends well to virtual three-tier design, which eliminates the need for a separate Application Server.
- **Cons**
 - Is generally more expensive than a two-tier and three-tier design since a separate Application Server and Workgroup Servers must be purchased. A Workgroup Server is needed for each remote LAN in the organization.
 - The performance over a WAN may still not be acceptable since transactional (non-static) data traffic must still travel over the WAN link.
 - Complexity is added since all Workgroup Servers must be kept in synchronization with the Master Database through data replication tools.
 - Client packages must be delivered over the WAN, which makes package management more complex. A remote file server may be needed to make the package available to all workstations on the remote LAN. A separate OCM map may be needed to direct the workstations to the static tables on the Workgroup Server.

5.2.2 Lite client

The lite client (also known as thin client) solution is the optimal WAN solution with very low WAN traffic since only presentation images are transmitted to remote workstations. It offers good security and manageability with all sensitive data and servers located at the company's central site.

Presentation functionality may be reduced slightly since an Internet browser is used for the user interface. OneWorld application development is not supported.

The lite client solution comes in several forms:

- OneWorld HTML Application Server
- OneWorld JAVA Application Server
- Microsoft (Citrix) TSE (Terminal Server Edition)

Figure 23 shows an example of a lite client design.

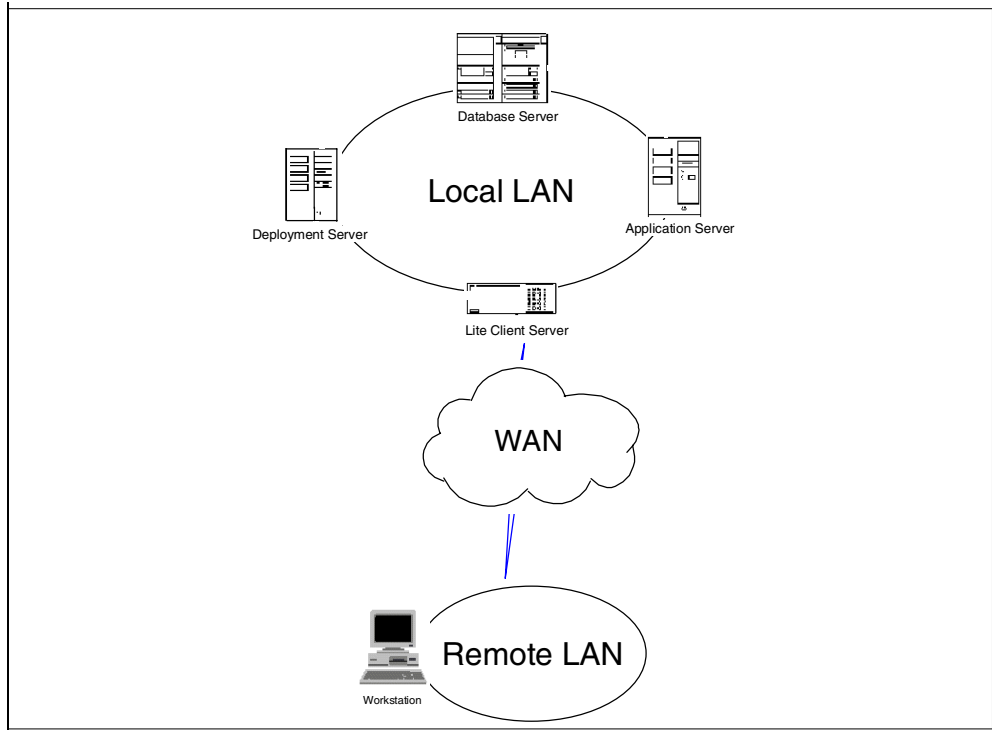


Figure 23. Lite client design

Using AS/400 system with a lite client

There are many variations of using the AS/400 system in this scenario. One of scenarios is shown in Figure 24.

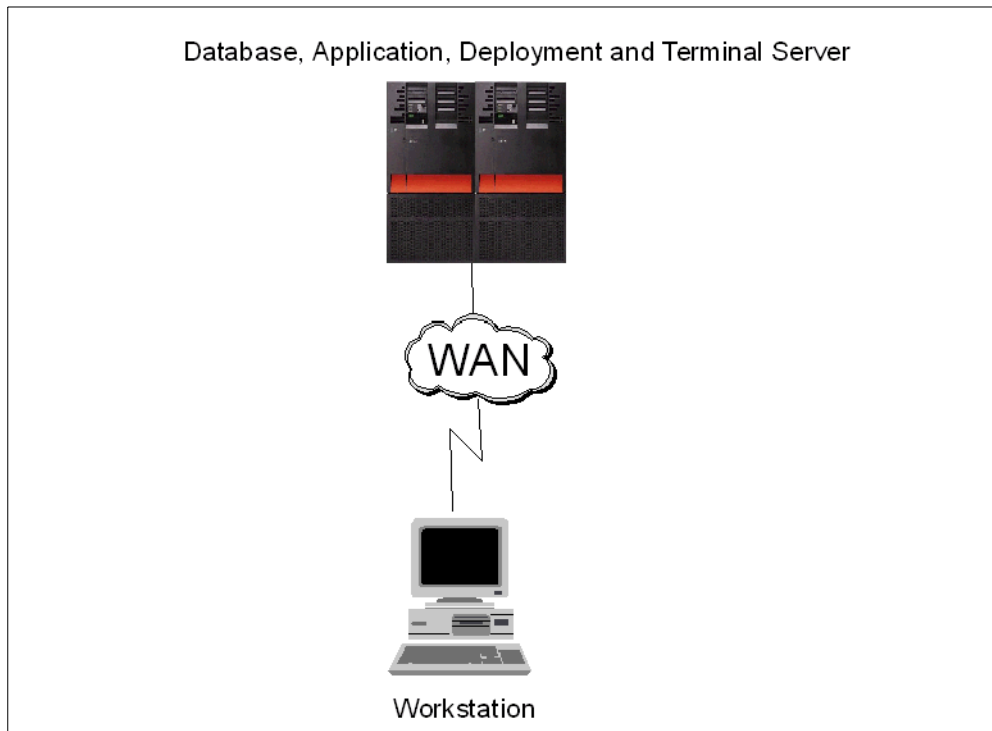


Figure 24. Using the AS/400 system with a lite client

In this scenario, the AS/400 system runs a Database Server, Application Server, Deployment Server, and Terminal Server, where the Deployment Server and Terminal Server run on one or more AS/400 Integrated Netfinity Servers. For more information on using Integrated Netfinity Server in this type of scenario, refer to 18.3.3, “Integrated Netfinity Server as a Deployment Server” on page 519, and 18.3.4, “Integrated Netfinity Server as Windows Terminal Server (WTS)” on page 521.

Pros and cons of using a lite client

This design has some important advantages over other designs described earlier:

- **Pros**

- Generally this is the best performing design for remote workstations on remote LANs since only presentation images are transmitted across the WAN. Data traffic over the WAN is virtually eliminated.
- Smaller and more diverse workstations can be used. The only requirement is that the remote workstations are able to run a Internet browser.
- Client package management is simplified since there is no need to deliver client packages to remote locations.
- Data security is better than the n-tier design since sensitive data is not duplicated to remote Workgroup Servers.

- **Cons**

- Remote application development is not supported.
- The presentation quality is lower since a Web browser is used for the workstation presentation.
- If a large number of remote workstations are supported, the cost of this design can rise rapidly. This is because of the need for multiple lite client servers. In general, a single, high-performance lite client server can handle 50 to 100 remote workstations. For installations with large numbers or remote workstations, a “server farm” may be required.
- If the Windows Terminal Server solution is used, products need to be purchased from Microsoft or Citrix.

5.2.3 Summary

It is important to understand that these OneWorld implementation designs are not mutually exclusive. On the contrary, most large installations will use a combination of them. For example, a company may choose to have all workstations on the local LAN use two-tier design, all remote developers use the three-tier or n-tier design, and all other remote workstations use the lite client design.

During the pre-installation analysis, a company should examine the needs of each employee group and decide on the design that will best serve those needs.

Chapter 6. Database

DB2 Universal Database for AS/400 (DB2 UDB for AS/400) is the integrated database management system (DBMS) of the AS/400 operating system that provides the structure, integrity, and data manipulation components of the relational database model. This chapter reviews the concepts and terminology of DB2 UDB for AS/400, SQL, query optimization tools and processes. This chapter also covers the OneWorld implementation of DB2 UDB for AS/400, including naming convention and methodology issues.

6.1 DB2 UDB for AS/400

DB2 UDB for AS/400 is the part of the operating system that controls the storing and accessing of data by an application program. The data may be on internal storage (for example, a database), on external media (diskette, tape, printer), or on another system. Data management provides the functions that an application uses in creating and accessing data on the system. It also ensures the integrity of the data according to the definitions of the application.

DB2 UDB for AS/400 is tightly integrated into the AS/400 operating system. In fact, it is impossible to operate an AS/400 system without using DB2 UDB for AS/400. OS/400 includes control language (CL) commands that perform many database functions. These include not only management commands but also operational commands.

Data management provides functions that allow you to manage files (create, change, override, or delete) using CL commands. The functions also allow you to create and access data through a set of operations (for example, read, write, open, or close). Data management also provides you with the ability to access external devices and control the use of their attributes for creating and accessing data.

If you want to make more efficient use of printers, data management provides the capability of spooling data for input or output. For example, data written to a printer can be held on an output queue until the printer is available for printing.

On the AS/400 system, each file (also called a table) has a description that describes the file characteristics and how the data associated with the file is organized into records. Whenever a file is processed, the operating system (Operating System/400 or OS/400) uses this description. Using the Display File Description (DSPFD) command for the file in question, you can review information and statistics on the file. The fields (also called *columns*) within a file can also be displayed using the DSPFD command.

You can create and access data on the system by using these file objects. Data management defines and controls several different types of files. Each file type has associated CL commands to create and change the file. You can also create and access data through the operations provided by data management.

The AS/400 system also implements a file management system similar to a PC file system called the integrated file system (IFS). For more information, refer to 2.8, "Integrated file system (IFS)" on page 17.

6.1.1 Data files

Data on the AS/400 system is stored in objects called *physical files* (in SQL terminology tables). An AS/400 physical file is located within an object called a *library* (*collection*). Physical files consist of *records* (rows) with a predefined layout. Defining the record layout means that you define the data structure of the physical file in terms of the attributes of data *fields* (columns) that make up that particular layout.

These definitions can be made through the native data definition language of DB2 UDB for AS/400 called *Data Description Specifications (DDS)*. Alternatively, the structure of a relational database can also be specified using the data definition statements provided by the Structured Query Language (SQL).

Using DDS, you can define *logical files* over your physical files or tables. Logical files provide a different view of the physical data, allowing column subsetting, record selection, joining multiple database files, and so on. They can also provide the physical files with an *access path* when you define a *keyed logical file*.

An *SQL view* is almost equivalent to an AS/400 native logical file. The selection criteria that you can apply in an SQL view is much more sophisticated than in a native logical file. An *SQL index* provides a keyed access path for the physical data exactly the same way as a keyed logical file does. Still, SQL views and indexes are treated differently from native logical files by DB2 UDB for AS/400 and cannot be considered to coincide exactly. For example, SQL views can have names longer than 10 bytes, but DB2 UDB for AS/400 files cannot.

For more information on DB2 UDB for AS/400 and SQL, refer to the *DB2 for AS/400 Database Programming Guide*, SC41-5701, *DB2 for AS/400 SQL Programming*, SC41-5611, and *DB2 for AS/400 SQL Reference Guide*, SC41-5612.

6.1.1.1 File types

The data management functions support the following types of files:

- **Database files:** Files whose associated data is physically stored in the system.
- **Distributed data management (DDM) files:** Allow access to data files stored on remote systems.
- **Intersystem communications function (OS/400-ICF) files** (hereafter referred to as ICF files): Allow a program on one system to communicate with a program on the same system or another system.
- **Save files:** Files that are used to store compressed data on disk (without requiring diskettes or tapes).
- **Device files:** Files that provide access to externally attached devices such as displays, printers, tapes, diskettes, and other systems that are attached by a communications line. The device files supported are display, printer, tape, and diskette files.

Each file type has its own set of unique characteristics that determine how the file can be used and what capabilities it can provide. However, the concept of a file is the same regardless of what type of file it is. When a file is used by a program, it is referred to by name, which identifies both the file description and, for some file types, the data itself.

A file can be used in many ways on the AS/400 system. For example, it can be accessed through a high-level language such as COBOL, RPG, C, or CL. Another way, which is becoming the standard in the computing industry, is through SQL. We cover the SQL basics in the following section

6.1.1.2 DB2 UDB for AS/400 latest enhancements

DB2 for AS/400 has long provided facilities to address most customer requirements. With V4R4, the AS/400 support has been extended to support new forms of information previously stored on the AS/400 system, but not managed by DB2 for AS/400. With this release, DB2 UDB for AS/400 now supports the storing, managing, and indexing of all forms of information including binary objects (such as spreadsheets, word processor documents, and multimedia objects) within the database. This support will include such features as binary large objects (BLOBs), user-defined functions, complex objects, query by image content, and even spatial extenders. All of these features allow customers to use one database management system to store, retrieve, and manage all of their corporate information.

Performance and functional enhancements to the DB2 Universal Database for AS/400 improve the processing of business intelligence queries. These improvements include:

- The hash “group by” algorithm improves performance of grouping queries for a large number of groups.
- The performance of MIN and MAX functions is improved with a suitable index, if available to determine the minimum or maximum value of a query.
- Derived tables and common table expression support allow complex business intelligence queries to be written without the use of views.
- Support for expressions in the GROUP BY and ORDER BY is also added.
- The QAQQINI file support allows the system administrator to set query specific options (query time limit and query degree) in a source file similar to QAQQINI file support used by PC applications. This aids in setting query options for client/server-based applications.

IBM provided the following enhancements to DB2 Universal Database for AS/400 through the 1999 Database Enhance Pack or Group PTF SF99014:

- **Large objects support:** With the addition of large objects (LOBs), DB2 Universal Database for AS/400 stores and manipulates data fields much larger than the current limits. An AS/400 record with LOB fields can hold up to 15 MB of data. With the new LOB support, DB2 UDB can be used as a platform for building applications that hold new non-traditional types of data such as image and audio as well as large text blocks.
- **DATALINK data type:** The DATALINK data type extends the types of data that can be stored in database files. The data stored in the column is only a pointer to the actual object such as an image file, a voice recording, or a text file. The method used for resolving to the object is to store a uniform resource locator (URL). This means that a row in a table can be used to contain information about the object in traditional data types, and the object itself can be referenced using the DATALINK data type.

DATALINKS also allow the referenced object to be “linked” to the database in such a way as to prevent the modification or deletion of the object while it is

linked to the database file. This relationship is maintained by having the database interact with the file system that contains the object.

- **User-defined data types:** User-defined data types are derived from existing predefined types such as integer and character data. You can create your own types for strong typing and creating functions for different types. You can call a function for each row of a result set and return a value based on the user-defined type.
- **User-defined data functions:** SQL now lets you define your own functions to use within the SQL itself. This saves you time in reusing common building blocks that you develop yourself. User-defined functions are necessary building blocks to support database extenders (extensions to support rich text and multimedia search and manipulation) currently supported on UDB.

6.2 AS/400 SQL concepts

SQL support on the AS/400 system consists of the following main parts:

- **SQL run-time support:** SQL run time parses and runs any SQL statement. This support is that part of the Operating System/400 (OS/400), and it does not require any other licensed program.
- **SQL precompilers:** SQL precompilers support precompiling embedded SQL statements in host programming languages. The following languages are supported:
 - ILE C/400
 - COBOL/400
 - ILE COBOL/400
 - AS/400 PL/I
 - RPG III (part of RPG/400)
 - ILE RPG/400

The SQL host language precompilers prepare an application program containing SQL statements. The host language compilers compile the precompiled host source programs. The precompiler support is part of the DB2 Query Manager and SQL Development Kit licensed program.

- **SQL interactive interface:** The SQL interactive interface allows you to create and run SQL statements. Interactive SQL is part of the DB2 Query Manager and SQL Development Kit for AS/400 (5769-ST1).
- **Run SQL statements CL command:** RUNSQLSTM allows you to run a series of SQL statements, which are stored in a source file. The Run SQL Statements (RUNSQLSTM) command is part of the DB2 Query Manager and SQL Development Kit licensed program.
- **DB2 Query Manager for AS/400:** DB2 Query Manager for AS/400 provides a prompt-driven interactive interface that allows you to create data, add data, maintain data, and run reports on the databases. Query Manager is part of the DB2 Query Manager and SQL Development Kit licensed program. For more information, refer to *DB2 for AS/400 Query Manager Use*, SC41-5212, and *DB2 for AS/400 Query Management Programming*, SC41-5703.
- **SQL REXX interface:** The SQL REXX interface allows you to run SQL statements in a REXX procedure. This interface is part of the DB2 Query Manager and SQL Development Kit licensed program.

- **SQL call level interface:** DB2 UDB for AS/400 supports the SQL Call Level Interface. This allows users of any of the ILE languages to access SQL functions directly through procedure calls to a service program provided by the system. Using the SQL Call Level Interface, you can perform all of the SQL functions without needing a precompile. This is a standard set of procedure calls to prepare SQL statements, execute SQL statements, and fetch rows of data. You can even perform advanced functions such as accessing the catalogs and binding program variables to output columns.

For a complete description of all the available functions, and their syntax, see *DB2 for AS/400 SQL Call Level Interface*, SC41-5806.

- **DB2 Multisystem:** This feature of the operating system allows your data to be distributed across multiple AS/400 systems. For more information on DB2 Multisystem, see the *DB2 Multisystem for AS/400*, SC41-5705.
- **DB2 Symmetric Multiprocessing (SMP):** This feature of the operating system allows the query optimizer to use parallel processing if the optimizer finds it appropriate. Symmetric multiprocessing is a form of parallel processing achieved by splitting a single job into multiple tasks, which are processed in parallel. One example of SMP usage is an n-way AS/400 system using multiple processors (either CPU or I/O processors) to process a job. Another example is a one-way AS/400 system that processes multiple tasks of the same job at the same time.
- **Application Program Interfaces (APIs):**

The following APIs are part of the OS/400:

- **QSQPRCED:** This API provides an extended dynamic SQL capability. SQL statements can be prepared into an SQL package and executed using this API. Statements prepared into a package by this API persist until the package or statement are explicitly dropped.
- **XDA family of APIs:** This family of APIs is built on QSQPRCED API. They provide additional capabilities handy in a distributed environment, such as remote program/system call and blocking and switching function. They also provide several optimizations to use QSQPRCED optimally and reduce communications traffic.
- **QSQCHKS:** This API syntax checks SQL statements.

For more information on these APIs, see the *System API Reference*, SC41-5801.

6.2.1 Terminology

In the relational model of data, all data is perceived as existing in tables. DB2 UDB for AS/400 objects are created and maintained as AS/400 system objects. Table 3 shows the relationship between AS/400 system terms and SQL relational database terms.

Table 3. Relationship of system terms to SQL terms

System terms	SQL terms
Library: Groups related objects. Allows you to find the objects by name.	Collection: Consists of a library, a journal, a journal receiver, an SQL catalog, and optionally a Data Dictionary. A collection groups related objects and allows you to find the objects by name.

Physical file: A set of records with fields.	Table: A set of rows with columns.
Record: A set of fields.	Row: The horizontal part of a table containing a serial set of columns.
Field: One or more bytes of related information of one data type.	Column: The vertical part of a table of one data type.
Logical file: A subset of fields and records of one or more physical files.	View: A subset of columns and rows of one or more tables.
SQL Package: An object type that is used to run SQL statements.	Package: An object type that is used to run SQL statements.
User Profile	Authorization name or Authorization ID

For more information on database, see *DB2 UDB for AS/400 Database Programming*, SC41-5701.

6.2.1.1 SQL terminology

There are two naming conventions that can be used in DB2 UDB for AS/400 programming: system (*SYS) and SQL (*SQL). The naming convention used affects the method for qualifying file and table names and the terms used on the interactive SQL displays. The naming convention used is selected by a parameter on the SQL commands or, for REXX, selected through the SET OPTION statement. The two naming conventions are further explained here:

- **System naming (*SYS):** In the system naming convention, files are qualified by library name in the form library/file.

Consider the case where the table name is not explicitly qualified and a default collection name is specified for the default relational database collection (DFTRDBCOL) parameter of the CRTSQLxxx or the Control SQL Package (CRTSQLPKG) commands. Then, the default collection name is used. If the table name is not explicitly qualified and the default collection name is not specified, the qualification rules are as follows:

- The following CREATE statements resolve to unqualified objects:
 - **CREATE TABLE:** The table is created in the current library (*CURLIB).
 - **CREATE VIEW:** The view is created in the first library referenced in the subselect.
 - **CREATE INDEX:** The index is created in the collection or library that contains the table on which the index is being built.
 - **CREATE PROCEDURE:** The procedure is created in the current library (*CURLIB).
- All other SQL statements cause the SQL to search the library list (*LIBL) for the unqualified table.

The default relational database collection (DFTRDBCOL) parameter applies only to static SQL statements.

- **SQL naming (*SQL):** In the SQL naming convention, tables are qualified by the collection name in the form collection.table.

If the table name is not explicitly qualified and the default collection name is specified in the default relational database collection (DFTRDBCOL)

parameter of the CRTSQLxxx command, the default collection name is used. If the table name is not explicitly qualified and the default collection name is not specified, the rules are:

- For static SQL, the default qualifier is the user profile of the program owner.
- For dynamic SQL or interactive SQL, the default qualifier is the user profile of the job running the statement.

Note that OneWorld uses the system naming convention for the AS/400 system.

6.2.2 Types of SQL statements

There are four basic types of SQL statements:

- Data definition language (DDL) statements listed in (Table 4).
- Data manipulation language (DML) statements (Table 4).
- Dynamic SQL statements (Table 5 on page 90).
- Miscellaneous statements (Table 5 on page 90).

SQL statements can operate on objects created by the SQL as well as on AS/400 externally described physical files and AS/400 single-format logical files, whether they reside in an SQL collection.

Table 4. DDL and DML SQL statements

SQL DDL statements	SQL DML statements
ALTER TABLE	CLOSE
COMMENT ON	COMMIT
CREATE COLLECTION	DECLARE CURSOR
CREATE INDEX	DELETE
CREATE PROCEDURE	FETCH
CREATE SCHEMA	INSERT
CREATE TABLE	LOCK TABLE
CREATE VIEW	OPEN
DROP COLLECTION	ROLLBACK
DROP INDEX	SELECT INTO
DROP PACKAGE	UPDATE
DROP PROCEDURE	
DROP SCHEMA	
DROP TABLE	
DROP VIEW	
GRANT PACKAGE	
GRANT PROCEDURE	
GRANT TABLE	
LABEL ON	
RENAME	
REVOKE PACKAGE	
REVOKE PROCEDURE	
REVOKE TABLE	

Table 5. Dynamic SQL and miscellaneous statements

Dynamic SQL statements	Miscellaneous statements
DESCRIBE EXECUTE EXECUTE IMMEDIATE PREPARE	BEGIN DECLARE SECTION CALL CONNECT DECLARE PROCEDURE DECLARE STATEMENT DECLARE VARIABLE DESCRIBE TABLE DISCONNECT END DECLARE SECTION INCLUDE RELEASE SET CONNECTION SET OPTION SET RESULT SETS SET TRANSACTION WHENEVER

6.3 DB2 UDB for AS/400 catalog views

This section describes the views contained in a DB2 UDB for AS/400 catalog. The database manager maintains a set of tables containing information about the data in the database. These tables are collectively known as the *catalog*. The catalog tables contain information about tables, parameters, procedures, packages, views, indexes, and constraints on the AS/400 system. The catalog tables include the following files in the QSYS library:

- QADBXREF
- QADBPKG
- QADBFDEP
- QADBXRDBD
- QADBFCST
- QADBCCST
- QADBIFLD
- QADBKFLD

The catalog tables also include the following files in the QSYS2 library:

- SQL_LANGUAGES
- SYSPARMS
- SYSPROCS

Unlike the other catalog tables in QSYS, these catalog tables cannot be populated from objects on the system. Therefore, you should regularly save these tables.

The database manager provides views over the catalog tables. The views provide more consistency with the catalog views of other IBM SQL products and with the catalog views of the ANSI and ISO standard (called *Information Schema* in the standard).

With the exception of SYSINDEXES, SYSKEYS, and SYSPACKAGES, each of the following catalog views has a corresponding view defined in the Information

Schema. The catalog includes the following views and tables in the QSYS2 library:

SQL_LANGUAGES	Information about the supported languages
SYSCHKCST	Information about check constraints
SYSCOLUMNS	Information about column attributes
SYSCST	Information about all constraints
SYSCSTCOL	Information about the columns referenced in a constraint
SYSCSTDEP	Information about constraint dependencies on tables
SYSINDEXES	Information about indexes
SYSKEYCST	Information about unique, primary, foreign keys
SYSKEYS	Information about index keys
SYSPACKAGE	Information about packages
SYSPARMS	Information about procedure parameters
SYSPROCS	Information about procedures
SYSREFCST	Information about referential constraints
SYSTABLES	Information about tables and views
SYSVIEWDEP	Information about view dependencies tables
SYSVIEWS	Information about the definition of a view

The catalog views in the QSYS2 library contain information about all tables, parameters, procedures, packages, views, indexes, and constraints on the AS/400 system.

In addition, an SQL collection contains a set of these views (except SQL_LANGUAGES, SYSPARMS, and SYSPROCS) that only contains information about tables, packages, views, indexes, and constraints in the collection.

Tables and views in the catalog are like any other database tables and views. If you have authorization, you can use SQL statements to look at data in the catalog views in the same way that you retrieve data from any other table in the AS/400 system. The database manager ensures that the catalog contains accurate descriptions of the objects in the database at all times.

When you are trying to view only a couple of columns, you notice that one of them is much longer than required. In this case, use the SUBSTR function on the SELECT statement to use a substring length of a column. The following SQL statement shows such a situation:

```
SELECT SUBSTR(NAME,1,30) , SYS_TNAME FROM HR750DMO/SYSTABLES
```

Note: The NAME column is 128 characters long, but most, if not all, names are less than 30.

This statement selects the NAME and SYS_TNAME columns from table SYSTABLES in collection HR750DMO. The two columns show up on the same display due to the substring function cutting off the unnecessary characters of the NAME column.

6.4 DB2 UDB for AS/400 query optimization

The optimizer, or the way in which your SQL statements are optimized, is a key factor in performance and the expectations of the response time for the users. A poorly written SQL statement, may involve additional overhead for the optimizer

to perform its function, causing increased response time for the users. In this case, a good database administrator that understands the optimization process is essential to a finely tuned and properly operating database.

6.4.1 General optimization strategy

The DB2 UDB for AS/400 Optimizer generally follows a particular process in determining the proper indexing to use. The steps involved with this process are:

1. Gather default statistics from the selection criteria and the tables being queried.
2. Generate the default cost.

The default cost is for the Non-Keyed Access method if an index is not required.

3. For each index (until timeout), perform the following actions:
 - a. Materialize the index attributes.
 - b. Determine if an index matches the selection criteria.
 - c. Update the selection criteria statistics.
 - d. Determine the cost of using the index.
 - e. Compare with the current best cost.

Starting with OS/400 V4R2, the optimizer starts looking at the indexes to consider in a logical manner rather than in the sequence of the last index created. The first index considered is the index that has the same fields as the fields in the where clause. Also, if the source of the SQL results is from ODBC, a timeout does not occur. All of the indexes are considered.

For V4R3, the user can set a value in the QQQOPTIONS data area to indicate ALLIO or FIRSTIO. The QQQOPTIONS data area only works for V4R3. Specifying an "F" forces optimization for the first screen, also known as *FIRSTIO. Specifying an "A" overrides to optimize for the entire query, also known as *ALLIO. For more information about these values, see the help text for the OPNQRYF CL command for the keyword OPTIMIZE for the values *ALLIO and *FIRSTIO. This override will override all query interfaces, such as OPNQRYF, SQL, STRSQL, STRQMQR, and so on. If an OPTIMIZE FOR nnnn ROWS clause is specified with the query, the optimization goal is not overridden. The same is true for the OPNQRYF when, for example, OPTIMIZE(*FIRSTIO 10) is specified. The value of "F" or "A" must be placed in the ninth byte of the data area. If you are not currently using a QQQOPTIONS data area, the following command can be used to create a QQQOPTIONS data area which specifies an "F":

```
CHGDTAARA DTAARA(LIBNAM/QQQOPTIONS (9 1)) VALUE('F')
```

This data area must be in the library list for the job before the first query operation for the job is performed. The query processing only checks for the existence of the QQQOPTIONS data area during the first query operations for a job.

For V4R4, the QQQOPTIONS data area has been replaced by the QAQQINI file. The QQQOPTIONS data area is no longer supported, and you *must* use the QAQQINI file. There is the option OPTIMIZATION_GOAL, which fulfills the same purpose as the QQQOPTIONS data area in V4R3. This option is currently undocumented but should be set to *FIRSTIO for OneWorld. A method to set this follows.

Here are the steps for forcing *FIRSTIO for SQL on the AS/400 system for V4R4:

1. Create a duplicate object of QSYS/QAQQINI into QUSRSYS including data.
2. Create a DFU over the first two fields (QQPARM and QQVAL) of QUSRSYS/QAQQINI.
3. Insert a new record with OPTIMIZATION_GOAL in QQPARM and *FIRSTIO in QQVAL.
4. Delete all nonsystem SQL packages.

For more information on the QAQQINI file, refer to *DB2 UDB for AS/400 SQL Programming*, SC41-5611, on the AS/400 Information Center at:

<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html>

6.4.2 Optimization tools

The AS/400 DB2 UDB for AS/400 has a sophisticated methodology used in optimization. With this in mind, certain tools are essential when dealing with analyzing the optimizer. The tools are needed to know what the queries and SQL you write are doing, or what the optimizer is doing with them. The following list shows the tools that you need to analyze your queries and the optimizer:

- Query optimizer debug messages
- Print SQL information
- Database monitor statistics
- Change query attributes

For a complete description of these tools, refer to Chapter 21, “Performance management” on page 561.

6.4.3 Encoded vector index (EVI)

EVI is a new system object type, designed explicitly for query workloads. EVIs were introduced in V4R3 to make dynamic bitmap indexing (introduced at V4R2) even more efficient on the AS/400 system. The original AS/400 indexing technology is based on a binary radix-tree-index structure. It has served DB2 well for years in the OLTP environment and in simple decision support solutions. When creating a keyed logical file or SQL index, the binary radix tree is the underlying structure used by OS/400. However, the heavy interest in business intelligence requires additional indexing technology tailored for the complex queries and large data stores associated with a data warehouse. EVIs complement the current radix tree indexes to provide fast response times to a wider range of queries on the AS/400 system.

EVIs are an advanced form of bitmap indexing that use a single array (or vector) to address some of the issues that arise when there are more than a few distinct key values (high cardinality). The EVI structure consists of a single vector and a symbol table. An example of an EVI structure is shown in Figure 25 on page 94.

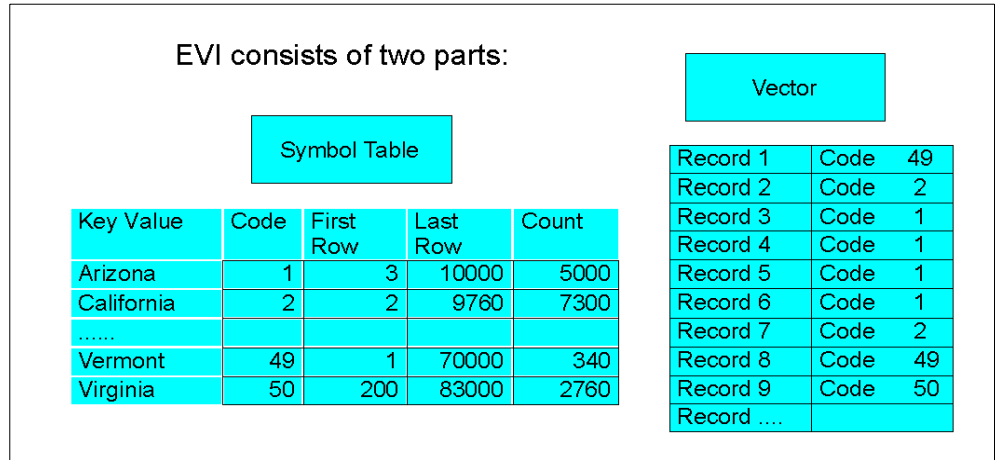


Figure 25. Encoded Vector Index structure

The symbol table contains an entry for each distinct key value and the unique code value assigned for that key value. Data statistics, such as the number of occurrences, are also maintained for each key value to aid in query optimization.

The single vector has an entry for each row in the base table. The entry contains the unique code value corresponding to the key value found in the row of the base table. In this example, California (code value 2) is the key value found in rows 2 and 7 of the table. The vector element is 1, 2, or 4 bytes in size depending on the number of distinct key values. If there are less than 256 distinct key values, a 1-byte element is used. If new distinct key values are added that overflow the number supported by the element size, DB2 UDB for AS/400 automatically rebuilds the EVI with a larger element size. In addition, the order of the elements in the vector matches the physical ordering of the table data, which leads to more efficient I/O processing.

6.5 Journaling

The main purpose of journal management on the AS/400 system is to enable you to recover the changes to a database file that occurred since the file was last saved. It is not necessary to run journaling on the AS/400 system to successfully use the database. This is different from both the Oracle and Windows NT SQL servers where journaling is integral to the implementation of the database.

You use a journal to define what files and access paths you want to protect with journal management. You may have more than one journal on your system. A journal may define protection for more than one file. However, you cannot journal a single table to more than one journal.

AS/400 system journaling uses two object types: journals (*JRN) and journal receivers (*JRNRCV). The journal acts like a funnel. All database table adds, changes, and deletes are received by the journal. The journal writes them to the journal receivers. The journal receivers are the keeper of these changes.

Note

For performance and availability reasons, we recommend that you put journal receivers on separate disk units, which can be achieved by using a user Auxiliary Storage Pool (ASP). See Appendix A, “Adding a user auxiliary storage pool” on page 775, to learn how to create a user ASP.

Journaling on the AS/400 system can be used for the following purposes:

- Commitment control
- Auditing
- Backup and recovery

The implementation of journaling is quite different based upon its intended usage.

If commitment control is implemented, it requires that the appropriate tables be journaled with both before and after images so that a rollback can be executed. If journaling is implemented for auditing or backup and recovery normally, only an after image is written. The choice of writing both images should not be affected by performance constraints since the actual write is buffered. The only difference is the amount of disk utilized.

If commitment control is implemented, the writes to the journal occur asynchronously since the database does not have to immediately know that the write was successful. However, at the end of a commit boundary, synchronous writes occur. If commitment control is implemented along with journaling, performance will be similar to a non journaled implementation.

If only journaling is implemented, the writes to the journal occur synchronously. This can cause a major degradation in performance. The same degradation also occurs when writes to a table occur outside a commitment boundary. Typically the best performance occurs when the commitment boundary spans approximately 1,000 records.

Journaling can have a major performance impact if the user implements journaling of all possible objects on the system to transmit to another system to provide a hot backup. In this case, remote journaling should definitely be implemented to reduce system overhead on both systems.

If journaling is heavily implemented, the MNGRCV(*SYSTEM) parameter should be used. This basically splits the journal receiver into two receivers: user data and system control information. This allows the system to now choose three additional drives in the ASP (allowing the maximum number used to increase from 10 to 13). When the system no longer needs its control information, it deletes it.

6.5.1 Introduction to remote journal function

The addition of the remote journal function on the OS/400 in V4R2 offers a reliable and fast method to transfer journal receiver data to a remote AS/400 system. It is ideal for use in data replication or high availability environments. Remote journal allows you to establish journals and journal receivers on the target system that are associated with specific journals and journal receivers on the source system. Once the remote journal function is activated, the source

system continuously replicates journal entries to the target system. The remote journal function is a part of the base OS/400 system and is not a separate product or feature. It is implemented at the SLIC layer. The benefits of the remote journal function include:

- It lowers the CPU consumption on the source machine by shifting the processing required to receive the journal entries from the source system to the target system.
- It eliminates the need to buffer journal entries to a temporary area before transmitting them from the source machine to the target machine. This translates into less disk writes and greater DASD efficiency on the source system.
- Since it is implemented in microcode, it significantly improves the replication performance of journal entries and allows database images to be sent to the target system in real time. This realtime operation is called the synchronous delivery mode. If the synchronous delivery mode is used, the journal entries are guaranteed to be in main storage on the target system prior to control being returned to the application on the source machine.
- It allows the journal receiver save and restore operations to be moved to the target system. This way, the resource utilization on the source machine can be reduced.

6.5.1.1 The traditional approach

Before describing the remote journal function, this section looks at the traditional approach for the high availability solution and its drawbacks. The high availability solutions, available prior to V4R2, used local journaling and the Receive Journal Entry (RCVJRNE) command. In a traditional environment, a user's applications that run on a source (production) system generate database changes. In turn, these changes create journal entries written to a local journal receiver. Still on the source side, the entries are received from the journal and buffered in a communications staging area. The data is transmitted asynchronously to the target system using existing communications gear. A high availability application running on the target system receives the journal entries into a temporary storage location, usually a user space. Another job, or many jobs, replay the changes into a copy of the source database. At this point, you have an exact copy of the production database on the target machine.

The asynchronous mode of operation means data latency. Your source system may fail while there are some journal entries waiting for transmission to the target machine. When this occurs, a few final database transactions are trapped on the source system. The database replica on the target system lags behind the production database on the source system.

The traditional approach also includes increased CPU utilization. The data passes through many layers of the system software. It crosses the Machine Interface (MI) boundary several times. This costs extra CPU cycles.

6.5.1.2 A remote journal solution

The remote journal function provides a much more efficient transport of journal entries than the traditional approach. In this scenario, when a user application makes changes to a database file, there is no need to buffer the resulting journal entries to a staging area on the production (source) system. Efficient low-level system code is used instead to capture and transmit journal entries directly from

the source system to associated journals and journal receivers on a target system. Much of the processing is done below the Machine Interface (MI).

Therefore, more CPU cycles are available on the production machine for other important tasks. Because the remote journal function, if activated in synchronous mode, replicates journal entries to the backup machine's main storage before updating the production's system database, the data latency is driven to zero. The high availability solutions available on the AS/400 system can fully take advantage of this more efficient transport mechanism. In fact, a high availability solution is still necessary in most customer environments to apply the Production (Source) System Backup (Target) System Journal application-dependent data to a replica database for hot-backup scenarios. It is also necessary for providing the required management facilities for these hot-backup environments. In addition, the existing high availability products support replication of objects other than database files.

6.5.1.3 Remote journal basics

When the remote journal function is activated on the source machine, the system replicates existing journal entries first as quickly as possible. This is referred to as *catch-up mode*. Once the specified journal receivers are transmitted to the target machine, the source starts continuously sending new entries either synchronously or asynchronously. The mode of operation depends on what was specified when the remote journal function was activated. The different delivery modes are discussed in the following sections.

You can consider the journal receivers on the target machine as a replica of the production machine's journal receivers. It is as if you saved the production machine's journal receivers and restored them on the target machine. The time stamps, system name, and qualified journal receiver names in the associated remote journal's journal entries are exactly the same as in the local journal's journal entries on the source system. In addition, the attach and detach times of the journal receivers are the same. However, while using remote journal support, you may see a minimal discrepancy in size between the local receiver and the associated remote receiver.

Please note that you are using two different machines, which pre-allocate space for the journal receivers in different operating Production System Backup System system environments. This may result in slightly different sizes, but the data in the associated receivers is always the same.

6.5.1.4 Synchronously maintained remote journals

In synchronous mode, journal entries are replicated to the main memory on the remote machine first. After an arrival confirmation is returned to the source machine, the journal entry is deposited to the local receiver. Next, the actual database update is made, if appropriate. The target system is updated in real time with all of the journal entries as they are generated by a user application on the source system. Synchronous journaling allows for recovery that loses no journal entries on the target system if an outage is experienced on the source system. Sending journal entries synchronously to a target system modestly impacts the journaling throughput on the source system. The main advantages of the synchronous delivery mode include:

- There are no trapped transactions on the production system. The synchronous mode should definitely be your choice if it is intolerable to switchover to the

target machine with a few of the final database transactions trapped on the source machine because these transactions were not sent to the target in real time.

- The journal image reaches the target machine before reaching the disk of the source system. Therefore, there is no delayed arrival of journal entries.
- The efficient utilization of high speed communications hardware (such as OptiConnect and ATM).

6.5.1.5 Asynchronously maintained remote journals

Sending journal entries asynchronously means that the journal entry is sent to the target system at some time after control is returned to the end user application that deposited the journal entry on the source system. From a recovery standpoint, asynchronous mode is less desirable. The reason is that the source system may have journal entries ahead of those journal entries that are known to the target system. Using this method allows for recovery that may lose a number of journal entries given a failure on the source system. It should have minimal impact to the local system when compared to the synchronous delivery mode. The main advantage of the asynchronous delivery mode is the minimal impact on the production machine.

The disadvantages of the asynchronous delivery mode include:

- A risk that some of the final database transactions may be trapped on the source system.
- In case of production machine's failure, the process of re-synchronizing the primary and replica databases is much more difficult when compared to the synchronous delivery mode.

However, asynchronous mode still remains a valid choice for those who intend to implement the remote journal function. You should consider asynchronous mode when:

- Your company can tolerate the delayed arrival of journal entries, such as in data warehousing environments.
- Your system is heavily utilized, and you do not have resources to compensate for even very moderate overhead of synchronous mode.
- You have slower communication gear that cannot keep up with the volume of journal traffic generated on your production machine.

6.6 J.D. Edwards database implementation

With the database as the heart and center of the J.D. Edwards application, it is important to understand how the database is implemented on the AS/400 system.

A complete database model does not exist for OneWorld. A limited set of table relationship diagrams for the major World tables exist both on the Web site and in printed form.

6.6.1 J.D. Edwards database naming conventions

The basic design of the OneWorld database originated with the design of J.D. Edwards WorldSoftware. This decision allowed for sharing the database between WorldSoftware and OneWorld software. Some of the newer applications have

either a different database design due to enhancements in the functionality or are completely new. However, the naming convention for the OneWorld tables and indexes is derived from the WorldSoftware naming convention.

6.6.1.1 Table naming convention

The naming convention used for database tables is shown in Figure 26.

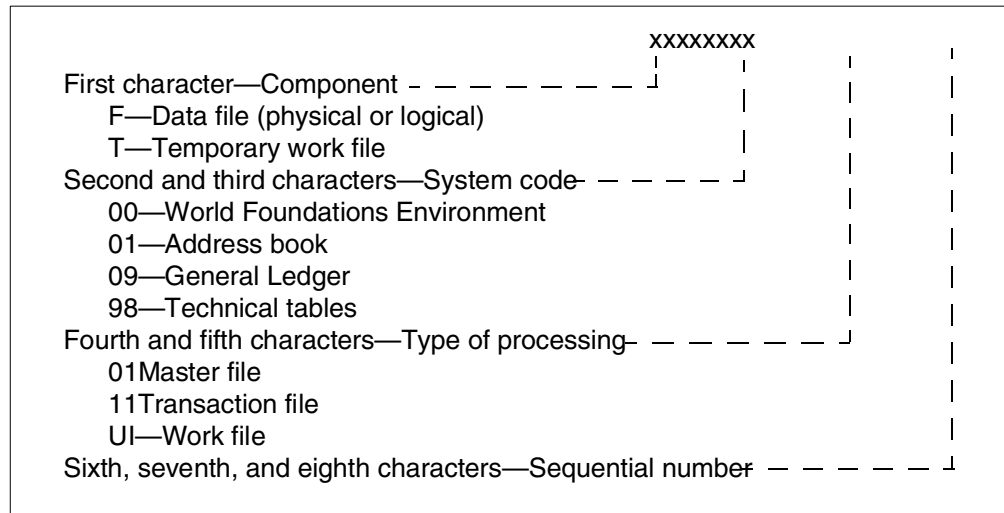


Figure 26. Table naming convention

For example, the address book master file would be named F0101, and the general ledger transaction file would be named F0911.

Work files do not always follow the convention of the fourth and fifth character being “UI”. Some of the old WorldSoftware work files end with a “W”. Another exception is that some system codes are three or four characters. All of the three-digit system codes have an alphabetic character as the third character.

For a list of both the system codes and the primary OneWorld tables, see the tips and technique document titled *OneWorld Table Definition* published by the Advanced Technologies group at J.D. Edwards. This document is intended to help the user analyze the placement of the tables in a distributed environment.

6.6.1.2 Indices

Most of the WorldSoftware physical files do not have an access path. However, some of the important system 00 files have an index on the physical file. The naming convention for WorldSoftware indices is to append an “L” (for logical file), followed by an alphabetic character starting with “A”. For example, the WorldSoftware logical file built over the address book number for the address book file would be named F0101LA. The “FxxxxLA” file normally is defined with a unique key constraint.

The WorldSoftware has relatively few joined files. The ones that exist are named by appending a “J” (for joined file) to the file name, followed by an alphabetic character starting with “A”.

OneWorld software names the indices by appending an underscore followed by a number starting with one to the table name. The primary key constraint required for other databases is the “_1” index. For example, the OneWorld index built over

the address book number for the address book file would be named F0101_1. This would also be the primary key. If the WorldSoftware file "F0101LA" exists, the OneWorld index "F0101_1" is not created. If the installation is not properly done, the OneWorld index "F0101_1" may exist as well as the WorldSoftware file "F0101LA".

The AS/400 system automatically attempts to share indexes. The attributes of the two indexes must be the same for this automatic sharing to take place. The primary criteria is the key fields must be the same (for a SQL created index) or a subset of the key fields (for a DDS created index). In addition to the key fields the index size, sequencing (unique, LIFO, FIFO, select/omit), language (CCSID, alt seq, etc), and finally the member specified for both indexes must be the same for the implicit sharing to take place.

OneWorld bases joins on the mapping of individual tables regardless of the location of the tables. The middleware (JDEBase) performs cross data-source join operations. This design allows cross platform implementation of the individual tables. If all of the joined tables are on the same machine, the middleware will create an SQL join statement. If the joined tables are not on the same machine, the middleware creates individual SQL select statements on each platform to create the functionality of the join. Use caution in planning the distribution of the tables so that associated tables that are used frequently do not cross platforms and impact performance.

The SQL implementation of OneWorld can be characterized as relatively simple. Over 80% of the SQL statements can be characterized as singular in structure. Relatively few JOINS occur in OneWorld. When JOINS occur, they usually only involve two tables. Common sense in table placement rarely causes a cross platform JOIN. This design tends to alleviate the need for EVI indexes. In some cases, building the EVIs are worthwhile for the statistics only, but this seems to be a rare situation.

6.6.1.3 Field naming convention

Just as a dictionary contains word definitions, the J.D. Edwards Data Dictionary is a central repository that contains data item definitions and attributes. The attributes determine how a data item:

- Appears on reports and forms
- Validates a data entry within an application
- Assigns column and row descriptions
- Provides text for field-sensitive help

The OneWorld Data Dictionary is active. Changes that you make are automatically reflected in applications without recompiling the software. The Data Dictionary tables are frequently replicated to the clients to improve performance.

If you are running a coexistence enterprise, you must create all of your data items in both WorldSoftware and OneWorld, because the two products cannot share the same Data Dictionary. However, a synchronization program is included to synchronize the WorldSoftware Data Dictionary to the OneWorld Data Dictionary.

The naming convention used for rows in the database tables is shown in Figure 27.

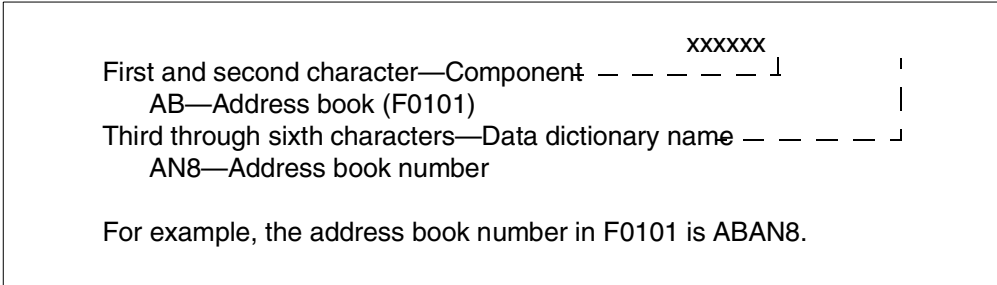


Figure 27. Naming convention used for rows in DB table

6.6.2 Coexistence versus OneWorld only

If the WorldSoftware is already loaded on the system and the installation process says to use these existing tables, OneWorld indexes are not created over the existing WorldSoftware tables. The installation process checks for the table's existence. If it exists, neither the table or corresponding indexes are created. The creation of the new indexes required by OneWorld is accomplished by the application of the WorldSoftware coexistence PTF and by the OneWorld installation process.

Implementation of the primary keys required by a OneWorld environment can result in problems in a few files. Although this rarely happens, a trial implementation needs to be scheduled to avoid the potential problem. The OneWorld environment was designed with a unique primary key requirement because other databases have this requirement. Since any table can be distributed to any supported database, this requirement had to be enforced by the database. The WorldSoftware environment does not enforce this requirement using the database. In WorldSoftware, records can be added to the database that are incomplete due to an abnormal failure. Since transaction processing is not implemented by WorldSoftware, these records remain in the database. This is typically the cause of the problem when converting between the two environments. Since the records should not be in the database originally, they usually can be simply deleted and the conversion process redone for that table.

6.6.3 Data sources

The AS/400 system implements an SQL table by referencing the LibraryName/TableName. In ODBC setup, this is called the system naming convention. Other databases implement an SQL table by OwnerName.TableName. On the AS/400 system, the owner of a table is not relevant. However, the library that contains a table is relevant. When ODBC points to an AS/400 data source, it normally points to a different AS/400 library for each data source.

There are a few tables in different default data sources that have the same name. The few tables in this status are basically the same, but the data is different. For simplicity, these tables should remain in separate libraries. For the majority of the tables, only two data sources can be used to reduce the number of jobs per user connection. The first data source has translation turned on, and the other data source has translation turned off.

6.6.4 Journaling and commitment control

OneWorld uses the AS/400 journaling and commitment control functions to implement Transaction Processing. Journaling on a table is dynamically started by OneWorld if OneWorld determines that this is necessary for the transaction processor. OneWorld defines the journal to be used by the following settings in the jde.ini file:

```
[DB SYSTEM SETTINGS]
Default Journal=
Default Journal LIBRARY=
Default Journal Receiver=
Default Journal Receiver LIBRARY=
```

Once OneWorld starts journaling on the AS/400 system, it is never ended.

Currently, OneWorld only implements transaction processing for a limited subset of the tables. Because of this, the performance tuning guideline for the placement of the journal and the journal receiver are not critical.

6.6.5 J.D. Edwards methodology

The key OneWorld methodologies that J.D. Edwards uses for the database implementation are discussed in this section.

6.6.5.1 Locking

In a heterogeneous network of database management systems, locking is handled in a variety of ways. Some database management systems use *row-level locking*, while some use *page-level locking*.

OneWorld does not implement any data-locking techniques. It relies on the native locking strategy of the vendor database management system. This improves performance by reducing duplication of efforts.

Row-level record locking can be turned on or off at the database data-source level. For more information about this topic, see "Data Sources" in the *Configurable Network Computing Implementation Guide*.

There are some specific situations when the vendor database does not automatically lock as needed. In these situations, you can use OneWorld data-locking. For example, you can use record locking to ensure the integrity of the Next Numbers facility.

OneWorld data-locking may be accomplished using one of the following methods:

- Two-phase commit
- Record locking
- Record is changed

Record locking

Record locking in OneWorld is implemented by calling published JDEBase APIs. Transaction processing uses a special set of locking APIs. When you use record locking, you should consider the time it takes to select and update a record, because the record is locked until the update is complete.

For example, suppose a user enters a transaction that uses Next Numbers. When they click OK, the Next Number function selects the appropriate Next Number

record, validates that this number is not already in the transaction file, and updates the Next Number record by incrementing the number. If another process tries to access the same Next Number record before the first process successfully updated the record, the Next Number function waits until the record is unlocked, and then completes the second process.

Record is changed

You can turn on the Lock Manager (Record change detection or Time stamp service) selectively by workstation. This type of database locking prevents a user from updating a record that changes during the time the user is inquiring on it. To change the record, the user must first re-inquire (reselect) the record and then make the change.

The following example illustrates a situation in which locking APIs are not used. For example, suppose two users are working within the Address Book application:

Time	Action
10:00	User A selects Address Book record "1001" to inspect it.
10:05	User B selects Address Book record "1001" to inspect it. Both users now have Address Book record "1001" open.
10:10	User B updates a field in Address Book record "1001", and clicks OK. OneWorld updates Address Book record "1001" with the update that User B made.
10:15	User A updates a field in Address Book record "1001", and clicks OK.

OneWorld does not update Address Book record "1001". The system displays a message informing User A that the record changed during the time User A was viewing it. For User A to change the record, User A must re-select it and perform the update.

When record change detection occurs, OneWorld displays the following message:

The record you are updating has been changed by another user. Please reselect your data.

To implement this function, a single server must be selected to run the Lock Manager. The jde.ini file for this server must have "AvailableService=TS". Each workstation that is to use the Lock Manager function must have the following code in the workstation jde.ini file:

```
[LOCK MANAGER}  
Server=server name  
RequestedService=TS
```

Note

The *RequestedService* value in the Enterprise Server jde.ini file should always be NONE.*

6.6.5.2 Transaction processing

Transaction processing ensures that related data is added to or deleted from the database simultaneously, therefore, preserving data integrity in your application.

In transaction processing, data is not written to the database until a commit command is issued. When this happens, all of the requests that are stored in the buffer are applied to the database at the same time.

For example, if a transaction comprises database operations to update two database tables, either all updates are made to both tables, or no updates are made to either table. This guarantees that the data remains in a consistent state and the integrity of the data is maintained.

You see a consistent view of the database during a transaction. You do not see changes from other users during a transaction.

Transaction processing ensures that transactions are:

- **Atomic:** Either all database changes for an entire transaction are completed, or the transaction is aborted and none of the changes happen.
- **Consistent:** Database changes transform from one consistent database state to another.
- **Isolated:** Transactions from concurrent applications do not interfere.
- **Durable:** Complete database operations are permanently written to the database.

Transaction

A OneWorld transaction is a logical unit of work (comprised of one or more SQL statements) performed on any number of databases.

You can construct a transaction within a OneWorld application to bind together multiple database operations. The application can request the database management system to buffer the database operations until the application executes a specific command to perform the updates requested within the transaction. Database operations that are not part of a transaction update the database immediately.

A single-statement transaction consists of one statement. A multiple-statement transaction consists of more than one statement.

Data interdependence

Data interdependence refers to the data elements that make a transaction complete. For example, a voucher has records in both the F0411 and F0911 tables. Because there is data interdependence between the two tables, the transaction is incomplete when there is data in one table and not the other.

Transaction boundaries

Data interdependence is defined by a transaction boundary. A transaction boundary encompasses all of the data elements that comprise a transaction. A transaction boundary may include only the data elements on a single form. When a transaction includes data from another form, the transaction boundary must be extended to include the data on that form.

Transaction processing and OneWorld

The OneWorld Application Design tool allows you to enable an application for transaction processing and to define what database operations comprise a transaction. Not all transactions or applications must be enabled. Enable transaction or applications appropriately according to your database configuration.

6.6.5.3 Commits

Transaction processing uses commits to control database operations. OneWorld applications use the auto or manual commit mode to update data. Commits are commands to the database. There are several types of commits, including:

- Commit
- Rollback
- Two-phase commit

There are two commit modes:

- Auto commit
- Manual commit

You use auto commit so that the system performs database changes immediately when an application submits an I/O request, and the system executes the request. Auto commit is the default.

Manual commit

When you use manual commit, database I/O requests are buffered until there is an explicit request from the application to commit or rollback the data. This type of commit ensures that, regardless of the time interval between the submission of statements, all the buffered statements update the database at the same time. This guarantees data consistency for statements that update dependent data. This mode is used with transaction processing.

Commit (manual commit mode)

A commit is an explicit command to the database to permanently store the results of operations performed by a statement. This is the successful end of a transaction.

Rollback (manual commit mode)

A rollback is an explicit command to the database to cancel the results of operations performed by a statement. This is the unsuccessful end of a transaction. In the case of a catastrophic failure (for example, network problems), the DBMS performs an auto-rollback. If the user clicks Cancel on a form, a rollback command is issued through a system function.

Two-phase commit (manual commit mode)

Two-phase commits coordinate a distributed transaction. They occur only when at least one update statement is executed to two separate data sources in the same transaction.

Transaction processing scenarios

Transaction processing is controlled by the application. It is enabled by checking the appropriate boxes in the form, the business function, the form interconnect, and in the table I/O.

Table 6 outlines the relationship between two forms and the boundaries that exist in each scenario. Transaction boundaries are defined through form interconnections and business function interconnections. In the following example, the OK button on Form 1 invokes Form 2. You can change the

transaction boundaries by changing TP On and TP Off. Table 6 explains what happens if you define your transaction boundary in various ways.

Table 6. Relationship between Form 1 and Form 2

Scenario	TP On	TP Off	Form, BSFN Interconnect, Table I/O	Comment
A	Form 1		X	All forms use auto commit.
	Form 2		X	
B	Form 1		X	Because neither form uses manual commit, the Include in Parent flag on Form Interconnect Properties is ignored.
	Form 2		X	All forms use Auto Commit
C	Form 1	X		Form 1 (parent) uses manual commit mode, and Form 2 (child) uses auto commit.
	Form 2		X	Because the Include in Parent flag is off, the transaction boundary does not extend to include Form 2 (child)
D	Form 1	X		Even though the transaction processing flag is off for Form 2 (child), the Include in parent flag is on.
	Form 2		X	The transaction boundary extends to include Form 2 (child).
E	Form 1		X	Because the Include in Parent flag is off, Form 1 (parent), and Form 2 (child) operate as independent entities.
	Form 2	X		Form 1 operates in auto commit mode, and Form 2 operated in manual commit mode.
F	Form 1		X	This is an odd case because transaction processing is off for Form 1 (parent). Therefore, the transaction boundary does not extend to the child, even though the Include in Parent flag is on for Form 2 (child).
	Form 2	X		Form 2 (child) is in manual commit mode, and the interconnect is ignored.
G	Form 1	X		Transaction processing is on for both forms.
	Form 2	X		Because the Include in Parent flag is off, each form is a transaction boundary and a commit is issued for each.
H	Form 1	X		Transaction processing is on for both forms. However, because the Include in Parent flag is on, the transaction processing on Form 2 is ignored.
	Form 2	X		The transaction boundary encompasses both forms. Form 2 is a child of Form 1.

6.6.5.4 Understanding transaction processing logging

The two-phase commit coordinator acts in two phases. In the first phase, it instructs the Log Manager to flush the logs for each data source to hard disk. This creates an application implemented recovery vehicle that can be used with any database. The logs contain every database operation that was carried out.

This action ensures that if any of the data sources fail to commit after the others committed, all databases can be returned to a consistent state by referring to the contents of the logs. If all the logs for each of the data sources are flushed successfully, the second phase begins.

In the second phase, the coordinator instructs each of the data sources to commit its respective transaction. If any of the data sources fails to commit, a commit log report is generated from the logs that were generated in phase one. This is written to the directory specified in the LOGPATH in jde.ini, which contains a listing by data source of all the SQL statements that were part of the transaction. The commit log also contains details as to which data sources passed and which ones did not.

This information helps the database administrator to manually synchronize the data sources so that they are all in a consistent state. A Recovery Manager can be implemented so that the recovery process does not have to be manual. The log report is only generated when at least one data source fails to commit. If all data sources successfully commit, no log report is generated. Then, all the logs from phase one are deleted by the Log Manager.

Chapter 7. Work management

Work management on the AS/400 system is the method used to manage system resources to achieve optimum throughput. The AS/400 system contains many objects that interact with each other and applications to process information efficiently. When the Operating System/400 (OS/400) licensed program is installed, it includes a work management environment that supports interactive and batch work. On the AS/400 client/server, work is considered batch work.

Client/server computing support on the AS/400 system is enabled by the OS/400 Host Servers option. The host servers interact with Client Access Express on a personal computer to provide access to AS/400 resources.

An understanding of the basic concepts of work management is an important prerequisite for understanding how the OneWorld client/server software runs on the AS/400 platform.

7.1 AS/400 work management concepts

This section discusses AS/400 work management concepts affecting job flow through the system. Since this section does not cover all aspects of the subject, refer to the *OS/400 Work Management Guide*, SC41-5306, for more information.

7.1.1 AS/400 system values

System values contain specifications that allow you to control or change the overall operation of your system. The system date is an example of a system value. A user can change the system value to define the working environment. For example, you can use the QDATFMT system value to specify the date format, such as YMD, MDY, DMY, or Julian format.

All available system values are categorized by the types that appear on the Work with System Values display, which you get when you use the WRKSYSVAL command. The types are:

- Allocation
- Date and Time
- Editing
- System Control
- Library List
- Message and Logging
- Security
- Storage

7.1.2 AS/400 network attributes

A network attribute is control information about the communications environment. Network attributes contain specifications that can be used for networking and communications. Network attributes are not objects and cannot be passed as parameter values like Control Language (CL) variables.

The AS/400 system is shipped with certain network attributes. Most of these are important only if your system is part of a communications network.

One important network attribute is the system name that appears on many OS/400 displays, including the sign-on display. Other network attributes include the local network ID, default mode for the system, network server domain, and many others.

You can view the current values of the network attributes by using the Display Network Attributes (DSPNETA) command. You can use the Change Network Attributes (CHGNETA) command to alter the values of any of the network attributes.

7.1.3 Subsystems

A *subsystem* is a single, predefined operating environment through which the AS/400 system coordinates the work flow and resource use. Subsystems are used to allocate main storage and provide a degree of isolation for jobs with similar processing characteristics (such as batch, interactive, and so on) in which to run. The AS/400 system includes several concurrently active subsystems. The separation of jobs into subsystems minimizes the contention for system resources and increases efficiency.

A subsystem description defines how work can enter the subsystem (work entries), how much work the subsystem can handle concurrently (activity level), how much main storage is allocated (pool specifications), and the relative importance of jobs in the subsystem (priority). The important components of a subsystem description are:

- **Operational attributes:** This is information such as the number of jobs that can be active in the subsystem at the same time and the interactive 5250-type sign-on display.
- **Storage pool definitions:** On the AS/400 system, all main storage can be assigned logically into units called storage pools. There are two types of pools:

- A *shared storage pool* is a pool in which multiple subsystems can run jobs. The system has two special shared pools and up to sixty-two general shared pools.

The special shared pools are the machine pool and the base pool. The *machine pool* (*MACHINE) is used for jobs that the system must run. The *base pool* (*BASE) contains all unassigned main storage that is not required by the machine storage pool or any other pool.

The general pools include the interactive pool, the spool pool and shared pools 1 through 60. The *interactive pool* (*INTERACT) contains assigned main storage that is used for interactive jobs, by default, unless specifically assigned to other main storage pools. The *spool pool* (*SPOOL) contains assigned main storage that is used by jobs that perform spooled file conversion, if necessary. For example, these jobs may perform Standard Character Set (SCS) to Advanced Function Printing Data Stream (AFPDS) conversion. The spool pool is also used by jobs that print spooled file output to local and remote (including network) printers. The additional shared pools (*SHRPOOL1 through *SHRPOOL60) can be optionally used and assigned main storage based on the anticipated number of subsystems and types of jobs that will run on the system.

- A *private storage pool* is an optionally assigned main storage pool that only supports jobs from a single subsystem. You can allocate as many as 62 private pools for use in active subsystems.

Storage pools also have an activity level and a paging option. The activity level is the maximum number of job threads that can be active at the same time in that pool. The paging option determines whether the system should dynamically adjust the paging characteristics of the storage pool for optimum performance. The four main storage pool attributes defined are:

- Pool identification (within the subsystem)
- Pool size
- Maximum activity level
- Paging option

- **Work entries:** A work entry specifies the source from which jobs can be accepted for processing in the subsystem. The work entry types are:
 - *Autostart job entries:* Each time the subsystem is started, the autostart jobs associated with it are started. This allows a one-time initialization of a job to perform a repetitive activity associated with the subsystem.
 - *Workstation entries:* A job is started when a workstation user signs on or when a workstation user transfers an interactive job from another subsystem. A workstation entry assigns a Non-Programmable Terminal (NPT) or workstation by type or by a specific device description name.
 - *Job queue entries:* Jobs are processed from the specified job queues. Job queue entries identify the job queues from which to take work and determine how much work to accept. Jobs are placed in a job queue using the AS/400 Submit Job (SBMJOB) command.
 - *Communications entries:* A communications job is a batch job that is started by a program start request from a remote system. For servers, the start request is initiated by a client or PC application. Communications work entries identify the sources from which the subsystem accepts start requests. A communications entry includes:
 - Device
 - Mode
 - Job description
 - Default user
 - Maximum active jobs

For example, a program start request using a mode entry of QSERVER is routed to the QSERVER subsystem. Users of other modes including QCASERV and QPCSUPP are routed to the QCMN subsystem.

- *Prestart job entries:* A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from other jobs because they use prestart job entries to determine which program, class, and storage pool to use when they are started. The objective of a prestart job is to perform as much “start-up” activity as possible before the remote request is received.
- **Routing Entries:** Routing entries specify the controlling program to be called to manage a routing step for a job running in the subsystem (typically the system-supplied program QCMD), which subsystem memory pool the job uses, and from which *class* to find the run-time attributes.

Routing data is the parameter used to select a routing entry for a job. Together, routing entries and routing data provide information on starting a job in a subsystem.

The class associated with a subsystem is an important object since it identifies many aspects of the job execution or run-time environment such as:

- *Run priority*: Run priority is a value, ranging from 1 (highest priority) through 99 (lowest priority) that represents the importance of the job when it competes with other jobs for the machine resources. This value represents the relative, not absolute, importance of the job. For example, a job with a priority of 25 is not twice as important as one with a priority of 50. This value is the highest run priority allowed for any thread within the job. Individual threads within the job may have a lower priority.
- *Time slice*: Specifies the maximum amount of processor time (in milliseconds) given to each thread in a job using this class before other threads in a job or other jobs are given the opportunity to run. The time slice establishes the amount of time allowed for the job to accomplish a meaningful amount of processing. If a thread exceeds its assigned time slice, it may be temporarily flagged as ineligible to run so that other threads can become active in the storage pool.
- *Eligible for purge*: Specifies whether a job is eligible to be moved out of main storage (main memory) and put into auxiliary storage (disk) at the end of a time slice or when there is a long wait (such as waiting for a workstation user's response). This attribute is ignored when more than one thread is active within the job. A job with multiple threads is never purged from main storage.
- *Default wait time*: Specifies the default maximum wait time (in seconds) that a thread in the job waits for a system instruction, such as the LOCK machine interface (MI) instruction, to acquire a resource. This default wait time is used when a wait time is not otherwise specified for a given situation. Normally, this is the amount of time the system user is willing to wait for the system before the request is ended.
- *Maximum CPU time*: Specifies the maximum processing unit time (in milliseconds) that a job can use. If the job consists of multiple routing steps, each routing step is allowed to use this amount of processing unit time. If the maximum time is exceeded, the job is ended.
- *Maximum temporary storage*: Specifies the maximum amount of temporary (auxiliary) storage (in kilobytes) that a job can use for processing. If the job consists of multiple routing steps, this is the maximum temporary storage that the routing step can use. This temporary storage is used for storage required by the program itself and by implicitly created.
- *System objects used to support the routing step*: If the maximum temporary storage is exceeded by a routing step, the routing step is ended. This parameter does not apply to the use of permanent storage, which is controlled through the user profile.
- *Maximum threads*: Specifies the maximum number of threads with which a job using this class can run, at any time. If multiple threads are initiated simultaneously, this value may be exceeded. If this maximum value is exceeded, the excess threads will be allowed to run to their normal

completion. Initiation of additional threads will be inhibited until the maximum number of threads in the job drops below this maximum value.

7.1.3.1 IBM supplied subsystems

Several subsystems are provided by and used with OS/400. IBM licensed programs and applications may place jobs into one of these subsystems or supply their own subsystems. This section gives a brief overview of key subsystems. For applications, you must review application specific documentation to determine their unique subsystem requirements.

IBM-supplied subsystems may use shipped system values (such as QCTLSBSD) and subsystem description parameters to determine the work or functions assigned to a particular subsystem. Through manipulation of the subsystem description, a user may route IBM provided applications to any subsystem or make adjustments to storage pools and job priorities. This section only describes the typical assignments.

Some of the subsystems included here have IBM-supplied job names for IBM supplied functions. Not all subsystems are listed, and not all jobs in every subsystem are listed.

Controlling subsystem (QBASE and QCTL)

The AS/400 system has one subsystem called the controlling subsystem. It is the first subsystem to start after an initial program load (IPL) and has to be active while the system is running. The system value QCTLSBSD specifies which subsystem is the controlling subsystem. The controlling subsystem is also the only subsystem active when the system is operating in a *restricted state*, which is needed when performing certain system functions such as complete system saves. The AS/400 system console runs in the controlling subsystem.

QBASE is the shipped default controlling subsystem and is typically used in simple, single application environments. Running non-interactive and interactive applications in QBASE at the same time generally results in page faulting rates that approach the poor range.

Most customers run a more sophisticated environment than can be supported by QBASE as controlling subsystem. Therefore, they set up QCTL as the controlling subsystem. In such case, the following subsystems are usually active instead of QBASE:

- QCTL as controlling subsystem
- QBATCH for batch-type work
- QINTER for interactive-type work

In some system environments, QCTL may also perform interactive and batch user application work. Other jobs, such as the system cleanup job QSYSSCD, may also run in QCTL.

While the controlling subsystem is started by AS/400 system, other subsystems are started either manually by a user or with a start-up program. The system value QSTRUPPGM specifies the name of this startup program that is called soon after the controlling subsystem is started at the end of an IPL. The startup program starts the subsystems that both IBM and the customer want to have active after an IPL. As shipped from IBM, the startup program starts the QBATCH, QINTER, QSPL, QSNADS, QCMN, and QSERVER subsystems.

Typically in a OneWorld environment, you would add the Start TCP/IP (STRTCP) and Start Host Server (STRHOSTSVR) commands to the startup program.

QBATCH and QINTER subsystems

Most OS/400 jobs that execute the Submit Job (SBMJOB) command defer the assignment of the job queue to the user profile's job description under which the job is running. The IBM-supplied default for this job queue, as defined in the job description, is QBATCH. The QBATCH job queue is assigned to the QBATCH subsystem. This means, by default, typical non-interactive jobs run in the QBATCH subsystem.

The QINTER subsystem is set up so that interactive sessions default to run in QINTER. This includes local and remote dependent workstation displays (5250 and 3270 displays), 5250 display station pass-through sessions, 3270-based sessions (such as DHCF and SNA Primary Logical Unit (SPLS)), or Client Access display emulation sessions, RUMBA/400 sessions, OS/2 Communication Manager 5250 emulation sessions, and ASCII Workstation Controller display devices.

QSPL subsystem

This subsystem is shipped to control all spooled printer output work.

QSERVER subsystem

The QSERVER subsystem runs the host server jobs for the Client Access *file serving* and *database serving* functions. The file server, the Database Server, and their associated daemon jobs *must* run in this subsystem. There is one autostart job, one file server job for each active client, and one Database Server job for an active database serving session, as shown in the following list.

In the TCP/IP environment, these jobs are:

- **User-id/QPWFSERVSO:** File serving support includes storing programs and files as a network drive (virtual disk) for the attached client.
- **User-id/QPWFSERVSD:** This is the file server daemon.
- **QUSER/QZDASOINIT:** There is one of these database serving functions for each active client session. QZDASOINIT is implemented as a prestarted job.
- **QUSER/QZDASRVSD:** This is the Database Server daemon.

In the SNA environment, these jobs are:

- **QPGMR/QSERVER:** This autostart job sets up the file serving and database serving environment on the AS/400 system.
- **User-id/QPWFSERV:** File serving support includes storing programs and files as a network drive (virtual disk) for the attached client.
- **QUSER/QZDAINIT:** There is one of these database serving functions for each active client session. QZDAINIT is implemented as a prestarted job.

QSYSWRK subsystem

The QSYSWRK subsystem is a common subsystem for various system jobs. Because of the possibility of a large number of different jobs active within this subsystem, it is important to understand what is currently known about these job types. For a particular customer environment, changes to the default run priority or storage pool assignment may be necessary to improve the overall system

performance. Additionally, all server jobs (except the file and Database Servers) run in this subsystem.

Subsystem description QSYSWRK is shipped to use only the base storage pool and is not included in the system-supplied IPL startup program QSTRUP. QSYSWRK is started by the SCPF job during an IPL unless the system performs IPL to a restricted state. A few of the more common jobs are:

- **Integrated Netfinity Server (formerly known as Integrated PC Server) monitor job:** There is one job active for each active Integrated Netfinity Server (INS) or Integrated PC Server (IPCS). The monitor job has the name of the network server description started for the INS or IPCS.
- **Mail Server Framework (QMSF/QMSF):** There can be one or more Mail Server Framework jobs (typically only one). The AS/400 system Start Mail Server Framework Job (STRMSF) command can be used to start multiple QMSF jobs. This may improve performance during periods of excessive sending and receiving of mail or SNADS distributions.
- **TCP/IP support:** TCP/IP support is included in OS/400. When the AS/400 system Start TCP/IP (STRTCP) command is issued, several jobs are started in QSYSWRK. These include the File Transfer Protocol (FTP) server, Telnet server, and Line Printer Daemon (LPD) server.

The QSYSWRK subsystem is shipped with several autostart job entries, including ones such as QSYSWRKJOB and QFSIOPJOB. These jobs run at the start of the QSYSWRK subsystem to perform such functions as QSYSWRK processing and restarting Integrated Netfinity Server (or Integrated PC Server) jobs.

QCMN Subsystem

The QCMN subsystem supports most communications jobs. User-written client/server application serving jobs (for example, using APPC or data queues) run in the QCMN subsystem. QCMN is active when the system value QCTLSBSD specifies the controlling subsystem as QCTL.

QSNADS subsystem

This subsystem usually performs functions such as document transmission and TCP/IP Simple Mail Transfer Protocol (SMTP) work over an SNA network. There can be several routing jobs active and a job for each send distribution defined for a remote location. A couple of the more common jobs are:

- **QSRVBAS/QESTP:** This job is activated as part of standard OS/400 support for receiving PTFs from IBM or a customer service provider.
- **QGATE/TCPIPLOC:** This job is activated when TCP/IP Simple Mail Transfer Protocol (SMTP) is activated for the local system.

7.1.3.2 Starting a subsystem

After a subsystem is started, the following sequence occurs:

1. The system interrogates the subsystem description for information.
2. Storage pools are allocated.
3. Display stations are allocated (sign-on displays are up) if appropriate.
4. Communications devices are allocated if appropriate.
5. Job queues are allocated.
6. Prestart jobs are started if appropriate.

7. Autostart jobs are started if appropriate.
8. The subsystem is up and running and ready for work.

7.1.4 Memory management

Memory management is a methodology that optimizes the use of installed main memory and improves the efficiency of the system. Some specific features of AS/400 memory management are explained in this section.

Memory Pools

The installed memory on the AS/400 system is partitioned into pools to minimize the contention for memory by jobs with different processing characteristics. For example, batch jobs are normally run in memory pools distinct from where interactive jobs run.

Activity Level

Activity Level determines the maximum number of jobs that may be active in the memory pool. An excessive value can result in increased page faulting due to many jobs competing for memory. A low value can result in jobs having to wait for an activity level to be freed.

Set Object Access (SETOBJACC) command

The Set Object Access (SETOBJACC) command allows information to be preloaded into a specified shared memory pool or private memory pool. The information can be programs or database files. This eliminates accessing disks for the preloaded objects. A good knowledge of the applications and database is a prerequisite for effective use of this facility.

Expert Cache

This is a selectable option under OS/400 that enables the system's single-level storage support to use main memory as a cache. Expert Cache is designed to reduce the number of physical disk I/Os but does not require a detailed understanding of the applications or database to be implemented. The operating system determines which objects (or portions of objects) are to remain in the shared storage pool where Expert Cache is enabled.

7.1.5 Jobs

An AS/400 job is a unit of work on the AS/400 system that is performed under a unique, fully-qualified name within the system. All jobs run within a subsystem.

7.1.5.1 Threads

A job consists of one or more *threads*. Each job has an *initial thread*. The initial thread is created when the job starts. The job may also have additional threads, identified as *secondary threads*, depending on the application.

A thread is an independent unit of dispatchable work. Each thread has its own execution environment, such as a call stack. The thread shares many of the resources that are assigned to the job. The identifier for the thread is unique in the job to which the thread belongs.

Controlling work in the system is performed mainly at the job level. Most commands and Application Programming Interfaces (APIs) operate against the entire job.

Each thread is an independent unit of work. The activity level of a storage pool applies to threads rather than jobs. However, in the subsystem description, maximum active counts associated with a subsystem and the subsystem work entries apply to jobs. Therefore, a thread is used in information about storage pool activity levels. A job is used in information about subsystem maximum active counts.

7.1.5.2 Types of jobs

The AS/400 system supports many different types of jobs. Here is a short summary of them:

- **Interactive:** Starts when a user signs on to the AS/400 system from an AS/400 display and ends when they sign off. The user interacts with the system by issuing commands, using function keys, and running programs and applications.
- **Group:** One of up to 16 interactive jobs that are associated in a group with the same workstation device and user.
- **Batch:** Has no interaction with the user and occurs when a user submits a job to a job queue, issues a communications program start request, starts a subsystem with an autostart job entry, or starts a subsystem with a prestart job entry. Compiling programs and reports are commonly run in batch.
- **Batch Immediate:** Is initiated (or spawned) directly by another job without going through any job queue. These jobs are like batch jobs in that they can exist without any direct communications with a user. Always run batch immediate jobs in the same subsystem as the job that started it.
- **Autostart:** A batch job that does repetitive work or one-time initialization work that is associated with a subsystem and is automatically started each time the subsystem is started.
- **Communications:** A batch job that is started by a program start request from a remote system.
- **Prestart:** A batch job that starts running before the remote program sends a program start request.
- **System:** A batch job created by OS/400 to control system resources and schedule jobs.

7.1.5.3 Job names

To make it easier to control and identify jobs on the system, each job has a unique, qualified job name. The qualified job name consists of three parts: the job name (or simple job name), the user name, and the job number.

For interactive jobs, the *job name* is the same as the name of the workstation the user is signed on to. The job name for a batch job can be specified up to 10 characters long.

The *user name* is the name of the user profile under which the job is started. For interactive jobs, the user name is the name entered in the user field on the sign-on display. For batch jobs, it is the user profile under which the batch job was started. The user name can be up to 10 characters long.

The *job number* is a unique number assigned by the system so users can identify jobs, even if more than one has the same job name and user name. The job number is always six numeric digits.

The syntax for qualified job names is similar to qualified names for objects in libraries. For example, if the job name is DSP01, the user is QPGMR, the job number is 000578, and the qualified job name is 000578/QPGMR/DSP01.

7.1.5.4 Job description

Job attributes determine how each job is run on the system. A subsystem knows where and when to get job attributes from a job description, user profile, system value, or the currently active job based on how each attribute is specified.

A job description contains a specific set of job-related attributes and can be used by multiple jobs. Therefore, when using a job description, a user does not need to specify the same parameters repeatedly for each job. Different job descriptions can be created to describe batch jobs or interactive jobs, or for each user of the system. Here is a partial list of job attributes:

- Job queue
- Job priority on job queue
- Output priority on output queue
- Print device
- Output queue
- Initial library list
- Message logging

7.1.5.5 Job execution

The following basic elements control the selection of active jobs for processing by the AS/400 system:

- **Task dispatching and priority:** The AS/400 operating system manages the multiple tasks to be processed through:
 - *Task Dispatch Queue:* Regardless of the number of CPUs in the AS/400 system, there is only one task dispatcher and Task Dispatch Queue (TDQ). All active system and user jobs are represented in this queue as Task Dispatch Entries (TDE) and basically are ordered by job priority.
 - *Job priority:* Mainly determines the order of jobs in the TDQ. However, there are other considerations that affect the position of jobs of equal priority in the TDQ. Jobs with a lower numeric value have a higher priority in the TDQ and are processed ahead of those with a lower priority, such as batch jobs.
- **Activity levels:** Each thread in a job must occupy an activity level to be processed by the CPU or CPUs. An activity level is specified to control the maximum number of jobs or threads that may be run concurrently and is specified at the system, subsystem, or memory pool level.
- **Time slice:** Is specified in the class and determines the amount of CPU seconds that each thread in a job is allowed to use to complete a task.
- **Job states:** A currently executing thread is in one of the following states:
 - *Active:* Occupying an activity level
 - *Waiting:* Waiting on an event to occur
 - *Ineligible:* Not entitled to use the processor

- **Job transitions:** As each thread in a job runs in the system, it is transferred between states. The following transitions can be displayed by using the AS/400 system Work with System Status (WRKSYSSTS) command and pressing the F11 key (Figure 28):
 - *Active-wait (A-W):* Once a thread in a job uses the CPU and the required task is completed, the thread enters a *wait* state until an event (such as the user pressing the Enter key or a function key in an interactive job) occurs.
 - *Wait-ineligible (W-I):* If the event, upon which the thread in a job was waiting, completes but an activity level is not available, the thread in the job enters an *ineligible* state.
 - *Active-ineligible (A-I):* If the thread in the job does not complete in the assigned time slice, it may become ineligible (if there are other threads or jobs waiting).

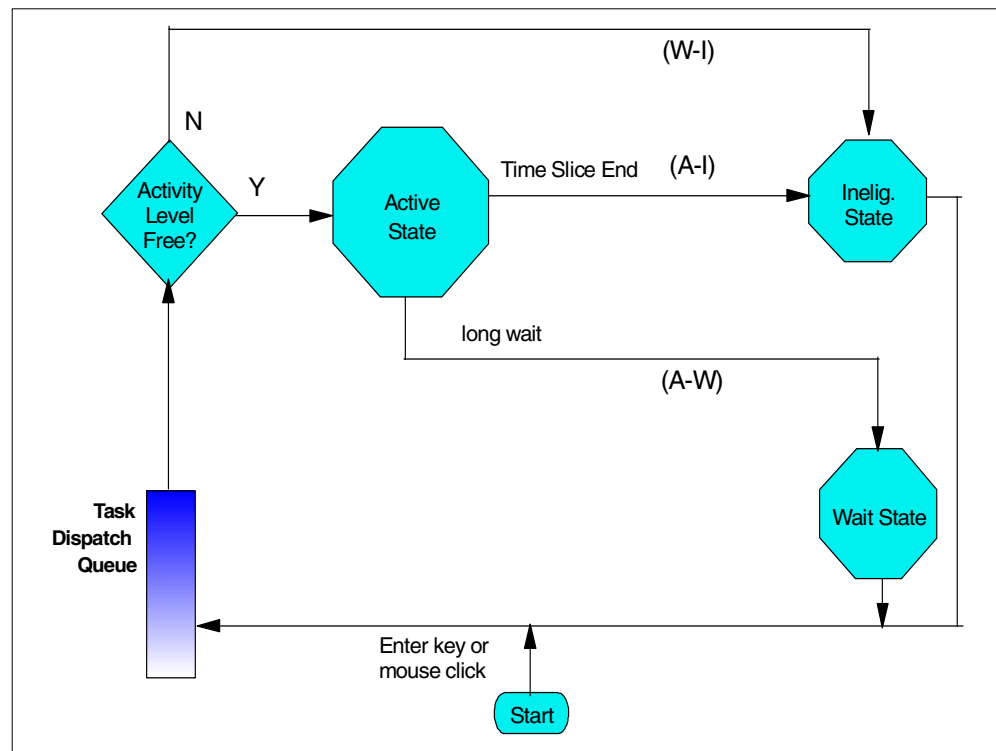


Figure 28. Thread transition states using the WRKSYSSTS command

7.1.6 Work Control Block Table (WCBT)

WCBT is a system object that maintains information for all jobs from the time they enter the system until they are removed. Each job is represented by a single entry in the WCBT.

Starting with V4R4, an index is maintained over the WCBT. This allows the system to locate entries much faster than on releases prior to V4R1 that use a sequential search algorithm. The WCBT consists of a single space that contains the header information and one to ten spaces of entries for jobs.

There is a Work Control Block Table Entry (WCBTE) for every job tracked by OS/400. Jobs are counted if they are:

- On the job queue ready to run
- Actively in the system doing work
- Completed with output that remains on an output queue

When a job is no longer tracked by the system, a new job that is starting can reuse its WCBTE. However, there can be situations on heavily used systems where the number of jobs being tracked is quite large. This causes the total storage occupied by the WCBTEs to increase.

If a large number of these entries (jobs) becomes “no longer tracked”, it can take a while to reuse the “available” entry space. During that time, processing the Work Control Block Table entries with such commands as Work With Subsystem Jobs (WRKSBSJOB) or the List Job (QUSLJOB) API can become time consuming on systems with a large number of tracked jobs. Eventually, you need to compress these entries to remove the empty slots to return to the count of entries specified with the QTOTJOB system value.

The compression is performed during IPL. Compression is requested using the Compress job tables (CPRJOBTL) parameter on the Change IPL Attributes (CHGIPLA) command. Compression reduces the size of the WCBT by freeing up the unused WCBTEs.

7.2 Client/server work management

In an AS/400 client/server computing environment, the OS/400 host servers option (OS/400 option 12) provides server side communications management. Client requests are provided by the IBM Client Access Express licensed program (see Chapter 11, “Client Access Express and OneWorld” on page 283). The support provided to Client Access Express is called *optimized support* to distinguish it from the support provided to PC Support/400 clients (*original support*). Only Client Access Express is supported with OneWorld.

The optimized servers that are provided with the OS/400 host server option in V3R1 and later releases use OS/400 sockets support to communicate with clients. In OneWorld, all communications between the client and host server take place using sockets.

Table 7 lists the programs used by the Client Access. More information is available in *Client Access Express Host Servers*, SC41-5740.

Table 7. Host servers for clients using sockets communications support

Server	Program name
File	QSYS/QPWFSEVSD - File Server daemon QSYS/QPWFSEVSO - Main server for each user
Secure File	QSYS/QPWFSEVSS - Secure File Server daemon
Database	QIWS/QZDASRVSD - Database Server daemon QIWS/QZDASOINIT - All DB Server requests
Secure Database	QIWS/QZDASSINIT - All Secure DB Server requests
Data Queue	QIWS/QZHQSRRVD - Data Queue Server Daemon QIWS/QZHQSRRV
Network Print	QIWS/QNPSEVSD - Network Printer Server daemon QIWS/QNPSEVSV
Central	QIWS/QZSCSRVSD - Central Server daemon QIWS/QZSCSRVS
Rmt Cmd/Pgm Call	QIWS/QZRCSRVD - Rmt cmd/Pgm call daemon QIWS/QZRCSRVS
Signon	QIWS/QZSOSGND - Signon Server daemon QIWS/QZSOSIGN

7.2.1 Optimized host servers

The optimized servers include:

- A file server that integrates with the integrated file system and allows clients to store and access information, such as files and programs, located on the AS/400 system
- A Database Server for Data Transfer, ODBC, Operations Navigator database, SQL APIs (DB APIs), and the Client Access Express OLE DB provider
- A network print server that allows enhanced client control over print resources on the AS/400 system
- A data queue server
- A remote command and program call server that allows PC applications to issue commands and call programs on the AS/400 system and return the results to the client
- A central server that provides services such as license management and other client management functions
- A signon server that provides password management functions for host servers with sockets support

7.2.2 Establishing client/server communications

Figure 29 on page 122 shows how client/server communications are established.

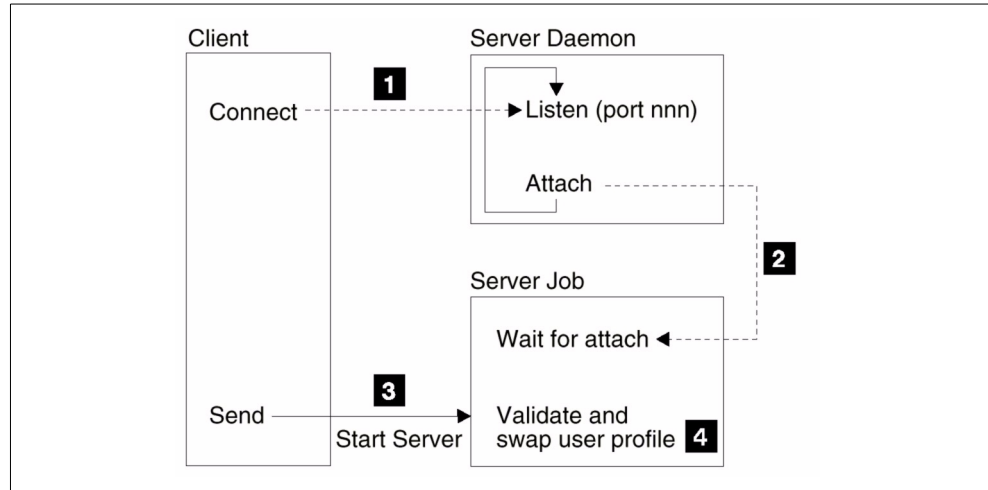


Figure 29. Establishing client/server communications

To initiate a server job that uses sockets communications support, the following process occurs:

Note: The numbers in black correspond to the numbers in Figure 29.

1. The client system connects to a particular server's port number **1**. A server daemon must be started (using the STRHOSTSVR command) to listen for and accept the client's connection request.
2. Upon accepting the connection request, the server daemon issues an internal request to attach the client's connection to a server job **2**. This server job may be a prestarted job or, if prestart jobs are not used, a batch job that is submitted when the client connection request is processed. The server job handles any further communications with the client.
3. The server connects to the client. The initial data exchange includes a request that identifies the user profile and password that are associated with the client user **3**.
4. Once the user profile and password are validated, the server job switches to this user profile, such as accounting code and output queue **4**.

For additional information about optimized servers and managing AS/400 servers with sockets communications, refer to *AS/400 Client Access Express Host Servers*, SC41-5740.

7.2.2.1 Server daemons

The server daemon is a batch job that is associated with a particular server type. There is only one server daemon per server type. However, one server daemon can have many server jobs.

The server daemon allows client applications to start communications with a host server that is using sockets communications support. The server daemon does this by handling and routing incoming connection requests. Once the client establishes communications with the server job, there is no further association between the client and the server daemon for the duration of that server job.

Start Host Server (STRHOSTSVR) command starts server daemons. The End Host Server (ENDHOSTSVR) command ends them. The server daemons must be active for the client applications to establish a connection with a host server that is using sockets communications support.

All of the server daemons run in the QSYSWRK subsystem, except for the database and file server daemon, which run in the QSERVER subsystem. The server daemon jobs run in the same subsystem as their corresponding server jobs.

The TCP/IP protocol and the associated subsystem must be active when the server daemon job is started.

7.2.2.2 Service table

The service table contains the port numbers of each server daemon and the symbolic service names. The service table is updated with these entries when Host Server option 12 is installed on the AS/400 system.

You can use the Work Service Table Entries (WRKSRVTBLE) command to see the service names and their associated port numbers. Table 8 shows the initial service table entries provided for the optimized servers.

Table 8. Port numbers for host servers TCP (Part 1 of 2)

Service name	Description	Port #
as-central	Central Server	8470
as-database	Database Server	8471
as-dtaq	Data queue server	8472
as-file	File server	8473
as-netprt	Network Print Server	8474
as-rmtcmd	Remote command/program call server	8475
as-signon	Signon server	8476

Table 9 shows the initial service table entries provided for the servers that use Secure Sockets Layer (SSL) support.

Table 9. Port numbers for host servers and server daemons using TCP (Part 2 of 2)

Service name	Description	Port #
as-central-s	Secure Central Server	9470
as-database-s	Secure Database Server	9471
as-dtaq-s	Secure Data queue server	9472
as-file-s	Secure File server	9473
as-netprt-s	Secure Network Print Server	9474
as-rmtcmd-s	Secure Remote command/program call server	9475
as-signon-s	Secure Signon server	9476

7.2.2.3 Starting and ending host servers

The Start Host Server (STRHOSTSVR) command starts the host server daemons. In addition, the STRHOSTSVR command attempts to start the prestart job that associates with the specified server types.

One server daemon exists for each of the host server types. The client application uses this port number to connect to the host server daemon. The server daemon accepts the incoming connection request and routes it to the server job for further processing.

The daemons are batch jobs that are submitted to either the QSYSWRK or the QSERVER subsystem. This depends on the value or values that are specified for the SERVER keyword. All daemon jobs are submitted to the QSYSWRK subsystem, with the exception of the *DATABASE and *FILE server daemons, which are submitted to the QSERVER subsystem. The server jobs run in the same subsystem as their corresponding server daemons.

To have the host servers start automatically when TCP/IP is started, complete the following steps by using Client Access Express:

1. Double-click the **Operations Navigator** icon.
2. If necessary, expand **My AS/400 Connections**.
3. Expand your AS/400 server.
4. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
5. Expand **Network**.
6. Expand **Servers**.
7. Click **Client Access**.
8. Right-click the host server (from Central, Database, Data queue, File, Net Print, Remote Command, or Sign On) that you want to start automatically, and select **Properties**.
9. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
10. Select the **Start when TCP/IP is started** check box on the General tab.
11. Click **OK**.
12. Now the jobs for that server will start when TCP/IP is started. Repeat steps 8 through 10 for each server to be started automatically.

Client Access servers support TCP/IP connections that use SSL. SSL encrypts all data that is sent between the client and server. To use SSL, you must have a certificate assigned to the host server. Certificates are assigned using digital certificate manager. For more information, go to the site at:

<http://publib.boulder.ibm.com/html/as400/infocenter.html>

Then, follow these steps:

1. Select **V4R4**. Select **English**. Click **GO**.
2. Click **Client Access Express**.
3. Click **Administering**.
4. Click **Connection Administration**.
5. Click **Secure Sockets Layer administration**.

See *Client Access Express for Windows - Setup V4R4M0*, SC41-5507, for information on how to establish SSL communications from the Client Access Express for Windows client.

The RQDPCL keyword is used to identify which communication protocols (TCP/IP or IPX) must be active at the time the STRHOSTSVR command is issued. If the required protocols are not active, the STRHOSTSVR command will fail. The server daemon can be started without any active protocols. When used in conjunction with the QZBSEVTM autostart job, the daemon job can dynamically identify when protocols become active.

Note: The host servers will not start if the QUSER password has expired. The password expiration interval should be set to *NOMAX for the QUSER profile. With this value, the password will not expire.

The End Host Server (ENDHOSTSVR) command ends the host server daemons. If the server daemon is ended, existing connections and jobs remain unaffected. However, subsequent requests from a client application will fail until the daemons are started again.

7.2.3 Client/server subsystems

The server jobs are configured to run in different subsystems depending on their function. The subsystems used for the server jobs are:

- **QSYSWRK subsystem:** All of the servers, with the exception of the database and file servers, the network drive server, and the transfer function server and their associated daemon jobs run in this subsystem. The file server and Database Server run in the QSERVER subsystem.
- **QUSRWRK subsystem:** This subsystem is where the server jobs run for the servers: Network Print, Remote Command/Program Call, Central, Data Queue, and Signon.
- **QSERVER subsystem:** The QSERVER subsystem runs the host server jobs for Client Access/400 *file serving* and *database serving* functions. The file server, the Database Server, and their associated daemon jobs must run in this subsystem. There is one autostart job and one file server job for each active client and one Database Server job for an active database serving session.

7.2.4 Prestart jobs

Prestart jobs are especially important in client/server communications. A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from other jobs because they use prestart job entries in the subsystem description to determine which program, class, and storage pool to use when the jobs are started.

Prestart jobs increase performance when you initiate a connection to a server. Using prestart jobs allows us to reduce the amount of time required to handle a program start request. The initiation of a prestart job follows this procedure:

1. When a subsystem is started, or when the Start Prestart Job (*STRPJ*) command is entered, prestart jobs are started based on the information contained in the prestart job entries.

2. When a program start request is received on the target system, it goes to the subsystem that has the required communications device allocated.
3. The program start request attaches to a prestart job that is already running if the subsystem finds either of the following entries:
 - A prestart job entry with a program name that matches the program name of a program start request
 - A routing entry that matches the routing data of the program start request and the routing program on the found routing entry matches the program name on a prestart job entry
4. If the prestart job entry is not active, the program start request is rejected. If a match is not found, the program start request causes a communications batch job to start if the routing data matches the routing entry. Otherwise, the program start request is rejected.

Prestart jobs can be reused, but there is no automatic cleanup for the prestart job once it is used and subsequently returned to the pool. The number of times the prestart job is reused is determined by the value specified for the maximum number of uses (MAXUSE) value of the ADDPJE or CHGPJE CL commands. This means that resources that are used by one user of the prestart job must be cleaned up before ending use of the prestart job. Otherwise, these resources will maintain the same status for the next user that uses the prestart job. For example, a file that is opened but never closed by one user of a prestart job remains open and available to the following user of the same prestart job.

By default, some of the server jobs run in QUSRWRK. QUSRWRK has prestart job entries added to it when the host servers option is installed. Using Operations Navigator, you can configure some or all of these servers to run in a subsystem other than QUSRWRK:

1. Double-click the **Operations Navigator** icon.
2. If necessary, expand **My AS/400 Connections**.
3. Expand your AS/400 server.
4. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
5. Expand **Network**.
6. Expand **Servers**.
7. Click **Client Access**.
8. Right-click the host server for which you want to configure subsystems and select **Server Jobs**.
9. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
10. Right-click the job that you want to change. Options that are available are shown in the drop-down box.
11. To move jobs from QUSRWRK to your own subsystem, you must:
 - a. Create your own subsystem description.
 - b. Add your own pre-start job using the ADDPJE command. Set the STRJOBS parameter to *YES.

If you do not do this, your jobs will run in QSYSWRK.

Table 10 and Table 11 summarize the prestart job default options shipped with OS/400 for TCP/IP connections.

Table 10. Default options for prestart jobs (Part 1 of 2)

Server	Network print	Rmt Cmd Pgm Call	Central	Database
Subsystem	QUSRWRK	QUSRWRK	QUSRWRK	QSERVER
Library/ program	QIWS/ QNPSEVS	QIWS/ QZRSRVS	QIWS/ QZSCRSVS	QIWS/ QZDASOINIT
User profile	QUSER	QUSER	QUSER	QUSER
Initial jobs	1	1	1	1
Threshold	1	1	1	1
Addl jobs	2	2	2	2
Max jobs	*NOMAX	*NOMAX	*NOMAX	*NOMAX
Max users	200	1	200	1
Wait for job	*YES	*YES	*YES	*YES
Pool ID	1	1	1	1
Library/class	QGPL/ QCASERVR	QGPL/ QCASERVR	QGPL/ QCASERVR	QSYS/ QPWFSEVER

Table 11. Default options for prestart jobs (Part 2 of 2)

Server	Secure database	File	Secure file	Data queue	Signon
Subsystem	QSERVER	QSERVER	QSERVER	QUSRWRK	QUSRWRK
Library/ program	QIWS/ QZDASSINIT	QSYS/ QPWFSEVERSO	QSYS/ QPWFSEVERSS	QIWS/ QZHQSSRV	QIWS/ QZSOSIGN
User profile	QUSER	QUSER	QUSER	QUSER	QUSER
Initial jobs	1	1	1	1	1
Threshold	1	1	1	1	1
Addl jobs	2	2	2	2	2
Max jobs	*NOMAX	*NOMAX	*NOMAX	*NOMAX	*NOMAX
Max users	200	*NOMAX	*NOMAX	200	200
Wait for job	*YES	*YES	*YES	*YES	*YES
Pool ID	1	1	1	1	1
Library/class	QSYS/ QPWFSEVER	QSYS/ QPWFSEVER	QSYS/ QPWFSEVER	QGPL/ QCASERVR	QGPL/ QCASERVR

The following list explains some of the terms used in Table 10 and Table 11:

- **Subsystem description:** The subsystem that contains the prestart job entries.
- **Library/program:** The program initiated by the prestart job.
- **User profile:** The user profile under which the job runs.
- **Initial Jobs:** The number of prestart jobs that are started when the subsystem is started.
- **Threshold:** The minimum number of currently unused prestart jobs that should remain active.
- **Additional jobs:** The number of additional jobs that are started when the number of jobs drops below the threshold value.
- **Maximum jobs:** The maximum number of jobs that may be active.
- **Maximum users:** The maximum number of times each prestart job is *reused* before it is ended.
- **Wait for job:** Causes a client connection request to wait for an available server job if the maximum number of jobs has been reached.
- **Pool ID:** The memory pool number within the subsystem that the job uses.
- **Library/class:** The execution class that the job uses.

When the start jobs value for the prestart job entry is set to *YES, and the remaining values are at their default settings, the following actions take place for each prestart job entry:

1. When the subsystem is started, one prestart job for each server is started.
2. When the first client connection request processes for a specific server, the initial job is used, and the threshold is exceeded.
3. Two additional jobs are started for that server based on the number that is defined in the prestart job entry.
4. The number of available jobs is always at least one, as specified in the threshold value of the prestart job entry.
5. The subsystem periodically checks the number of prestart jobs that are ready to process requests and ends excess jobs. The subsystem always leaves at least the number of prestart jobs specified in the threshold parameter.

7.2.4.1 Monitoring prestart jobs

Use the Display Active Prestart Jobs (`DSPEACTPJ`) command to monitor the prestart jobs. For example, to monitor prestart jobs for the signon server, you must know in which subsystem your prestart jobs are (QUSRWRK, QSYSWRK or QSERVER) and the program (for example, QZSOSIGN). Refer to Figure 30 and Figure 31 for more information.

```

                                Display Active Prestart Jobs
                                ASM23
                                02/19/99 12:24:58
Subsystem . . . . . : QSYSWRK      Reset date . . . . . : 02/10/99
Program . . . . . : QZSOSIGN      Reset time . . . . . : 09:24:44
Library . . . . . : QIWS          Elapsed time . . . . . : 0219:00:14

Prestart jobs:
Current number . . . . . : 10
Average number . . . . . : 8.5
Peak number . . . . . : 25

Prestart jobs in use:
Current number . . . . . : 5
Average number . . . . . : 4.3
Peak number . . . . . : 25

More...

Press Enter to continue.

```

Figure 30. Monitoring prestart jobs with DSPACTPJ (Part 1 of 2)

```

                                Display Active Prestart Jobs
                                ASM23
                                02/19/99 12:24:58
Subsystem . . . . . : QSYSWRK      Reset date . . . . . : 02/10/99
Program . . . . . : QZSOSIGN      Reset time . . . . . : 09:24:44
Library . . . . . : QIWS          Elapsed time . . . . . : 0219:00:14

Program start requests:
Current number waiting . . . . . : 0
Average number waiting . . . . . : 0.2
Peak number waiting . . . . . : 4
Average wait time . . . . . : 00:00:20.0
Number accepted . . . . . : 0
Number rejected . . . . . : 0

Bottom

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F13=Reset statistics

```

Figure 31. Monitoring prestart jobs with DSPACTPJ (Part 2 of 2)

7.2.4.2 Managing prestart jobs

The information about program start requests can indicate whether you need to change the available number of prestart jobs. If the information indicates that program start requests are waiting for an available prestart job, you can change prestart jobs by using the Change Prestart Job Entry (CHGPJE) command.

If the program start requests are not acted on quickly, you can perform any combination of the following options:

- Increase the threshold.
- Increase the parameter value for the initial number of jobs (INLJOBS).
- Increase the parameter value for the additional number of jobs (ADLJOBS).

The key is to ensure an available prestart job exists for every request.

For additional information about optimizing prestart job settings, refer to “Prestart job QZDASOINIT settings” on page 580.

7.2.5 Open Database Connectivity (ODBC) connections

Client side applications communicate with AS/400 server applications by using ODBC. ODBC is a Microsoft architected database access interface that enables applications to access data using Structured Query Language (SQL) as a standard language.

ODBC requests are submitted to the QSERVER subsystem where QZDASOINIT pre-started jobs run. Each request to connect a specific data source initiated by the Client Access/400 ODBC driver uses one QZDASOINIT pre-started job, servicing the user profile specified in the connection string. The programs QZSCSRVS and QZSOSIGN are also used for making the connection. For more information about OneWorld ODBC jobs, refer to “OneWorld ODBC database connection jobs” on page 726.

7.3 OneWorld work management

This section ties together AS/400 system work management concepts with the OneWorld environment.

7.3.1 OneWorld middleware

OneWorld middleware, which resides between the operating system and OneWorld business applications, provides the mechanism for bridging multiprotocol and multivendor communications issues. This architecture allows OneWorld to be based on a single, consistent communications platform.

OneWorld depends on two types of middleware: JDENet Communications Middleware and JDEBase Database Middleware. For more information on these two types, refer to 4.8, “Middleware” on page 65.

7.3.2 OneWorld process-based design

Only one instance of the OneWorld client can run on a particular OneWorld client workstation at any one time. When communicating with a OneWorld server, the OneWorld client uses dynamic-link libraries (DLLs) to run an internal network process. This process is the client-side portion of the JDENet communications middleware and is also referred to as a *net process*.

On the OneWorld server side, the JDENet process communicates with the OneWorld client and routes request messages to appropriate dedicated processes. In turn, the dedicated processes route work to the appropriate server platform-specific logic processes, such as DLLs, shared libraries, and job queues. A server can have multiple OneWorld main processes, multiple OneWorld dedicated processes, multiple DLLs, shared libraries, and job queues.

The advantage of this client/server architecture is that multiple workstations can make requests to the same server at one time. You can control the number of workstations that can make and maintain a session connection to a main server process. Furthermore, you can define the total number of dedicated processes (and the number of each type) that OneWorld uses to process specific types of workstation requests.

7.3.2.1 Network processes

Every Enterprise Server must have at least one OneWorld network process. The purpose of this constantly running job is to handle OneWorld communication messages by monitoring the network for incoming messages to OneWorld and also monitoring OneWorld for outgoing messages to the network. There can be multiple JDENET_n processes, but the initial JDENET_n process is the Master Listener. The Master Listener is the only process that directly monitors the network for OneWorld messages and passes those messages to additional JDENET_n processes, if defined, or to other jobs for processing as required.

If multiple JDENET_n jobs are specified, OneWorld starts the jobs as required, by allocating a job to each incoming request. When the maximum number of JDENET_n processes are started, OneWorld automatically alternates assigning the incoming requests between the currently running JDENET_n jobs until the maximum number of connections is reached.

7.3.2.2 Kernel processes

The JDENET_n process is only responsible for message handling. If the JDENET_n job determines that the incoming message is a request for logic processing, it routes the request to an appropriate JDENET_k job, called a *OneWorld kernel process*. The responsibility of the JDENET_k job is to handle the two-way routing to and from the various logic processes while the JDENET_n job handles the return delivery to the appropriate machine.

OneWorld Release B733.2 uses 12 predefined and dedicated kernel types, which are essentially virtual servers. Each kernel type is responsible for a specific OneWorld process.

OneWorld determines an appropriate JDENET_k job based on message identifiers generated by OneWorld. The JDENET_k job is the OneWorld process that provides the link between the JDENET_n job and the appropriate platform-specific processing job. The JDENET_k process only applies to servers.

Examples of logic processes include DLLs for Windows NT platforms, shared libraries (.sl) for UNIX platforms, and JDENet processes for AS/400 platforms.

7.3.2.3 OneWorld dedicated process design

Because JDENet is a message-based architecture, you can label each message with a message type identifier. JDENet uses this identifier to route messages to a range of processes dedicated to handling requests that fall within a specified range. The following process types are defined:

- Type 1** OneWorld internal and testing processes
- Type 2** OneWorld batch process (UBE) pass-through
- Type 3** Data replication requests
- Type 4** Security processes
- Type 5** Transaction manager and lock manager

- Type 6** Remote Master Business Function (MBF)
- Type 7** JDBNET Server-to-Server
- Type 8** Package Install
- Type 9** Server Administration Workbench (SAW)
- Type 10** Scheduler
- Type 11** Package Build
- Type 12** UBE System

7.3.2.4 Adding additional kernels when mapping MBFs

In a Windows Terminal Server setup, it is necessary to map Master Business Functions (MBFs) to the AS/400 system. Whenever MBFs are mapped to an AS/400 system, additional MBF kernels need to be set up to accommodate the increased workload. Use the guideline of 5 to 10 users per kernel. An example would be if Sales Order Entry MBFs are mapped to the AS/400 system, and your company has 30 Sales Order Entry users. Then, six MBFs kernels should suffice. Since each MBF kernel will consume about 40 Mb of RAM per kernel, consider the amount of available memory on your system before being too liberal with MBF kernels.

To increase the number of MBF Kernel processes, follow these steps:

1. Enter the following command:

```
WRKMBRPDM FILE(B7332SYS/INI)
```

2. Enter 2 to edit the JDE member:

- a. Locate the [JDENET_KERNEL_DEF6] section, and change the following value:

```
maxNumberOfProcesses="Total Number of MBF Kernel processes"
```

- b. Increase the [JDENET] section:

```
value maxKernelProcesses="Total of all the JDENET_KERNEL_DEF* section values, maxNumberOfProcesses=n"
```

7.3.2.5 Adding additional JDENET processes to service MBF kernels

When MBFs are being mapped to your Enterprise Server, it may be necessary to setup additional JDENET processes to service these MBF kernels. A guideline of one JDENET process for every 10 MBF kernels is good guideline. Therefore, in the example above for 30 Sales Order Entry users, a total of three JDENET processes would be necessary.

To increase the number of JDENET processes, follow these steps:

1. Type the following command:

```
WRKMBRPDM FILE(B7332SYS/INI)
```

2. Enter 2 to edit the JDE member

3. Locate the [JDENET_KERNEL_DEF1] section, and change the following value:

```
maxNumberOfProcesses="Total Number of JDENET processes"
```

4. Increase the [JDENET] section:

```
value maxKernelProcesses="Total of all the JDENET_KERNEL_DEF* maxNumberOfProcesses=n"
```


7.3.3 OneWorld AS/400 architecture and process flow

OneWorld architecture is explained in Chapter 3, “OneWorld internal architecture” on page 31.

All communications between the client and the host server take place using sockets. The communications between JDENET_N and JDENET_K occur with shared memory. JDENET_N and jdequeue communicate through the F986110 database table.

Figure 32 shows the OneWorld AS/400 architecture and process flow.

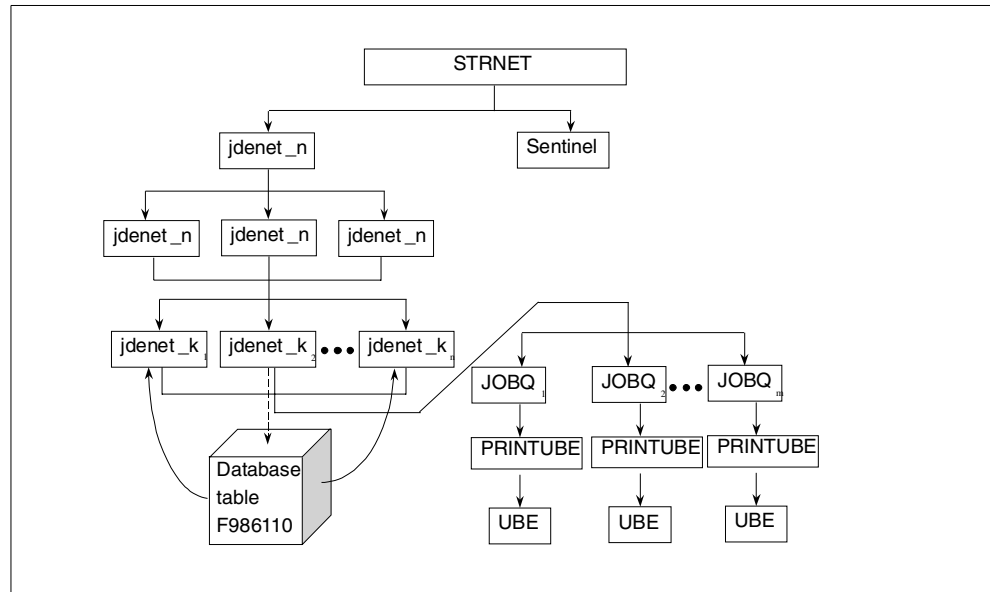


Figure 32. OneWorld AS/400 architecture and process flow

7.3.3.1 Process flow

AS/400 JDENET job startup follows this process:

1. Executing the OneWorld AS/400 STRNET command results in a single JDENET_N (NETWORK; the “N” stands for “Network”) job starting and a SENTINEL job starting in a newly started subsystem. The JDENET_N job handles the communications requests. The SENTINEL job monitors the number of JDB_x programs that are available and in use. The JDB_x programs are used to ensure commitment control is correctly scoped. There should always be a SENTINEL job running while OneWorld's JDENET is active.
2. Each time a UBE or CallObj (execute a Business Function) is submitted, another JDENET_N job is started in the same subsystem until the number of JDENET_N jobs equals the maxNetProcesses field value in the [JDENET] section of the INI file.
3. Each time a UBE or CallObj is submitted, a JDENET_N job starts another JDENET_K job until the number of JDENET_K jobs equals the maxKernelProcesses field value in the [JDENET] section of the INI file.

When a user submits a UBE or a server package installation, JDENET_N (as part of the host server) communicates with the client. The following actions occur:

1. The host server programs are B7332SYS/JDENET_N.
2. The client environment is initialized.
3. The client tells the host server (using a socket) to initialize its environment.
4. The host server (for example, JDENET_N) initializes its environment and gets environment and user handles.
5. The host server passes the environment and user handles to the client (using a socket).
6. The client launches the UBE or deploys the server package installation. It then sends data to the host server (using a socket).
7. If the maximum number of kernel (for example, JDENET_K; the "K" stands for "Kernel") jobs has not been met, JDENET_N starts a new JDENET_K job.
8. If the maximum number of JDENET_K jobs has been met, JDENET_N waits until a JDENET_K job is ready for more input.
9. The client frees the user environment.
10. The client tells the host server (using a socket) to free the server's user environment.
11. The host server frees its user environment.
12. The client tells the host server (using a socket) to free the server's environment.
13. The host server frees its environment.

JDENET_K (the kernel job) writes to the database (UBE or server package installation only). The following actions occur:

1. The program is B7332SYS/JDENET_K.
2. JDENET_K adds a record in the F986110 database table. The record has a status of "W" (Waiting).
3. JDENET_K submits a job to the job queue (JOBQ) associated with the job description (JOBQ) of the user submitting the UBE. The job queue name sent to the AS/400 Enterprise Server is ignored. This job calls the OneWorld program PRINTUBE on the AS/400 Enterprise Server.

The submitted PRINTUBE job executes the UBE or server package installation. The following actions occur:

1. The program is B7332SYS/PRINTUBE.
2. Changes the status stored in database table F986110 to "P" (Processing).
3. Starts the UBE or server package installation.
4. If the UBE or installation completes successfully, it changes the status in F986110 to "D" (Done).
5. If the UBE or installation fails, it changes the status in F986110 to "E" (Error).

As opposed to the many processes that execute when a UBE or server package installation is submitted, JDENET_K performs the processing when a user submits a CallObject command.

7.3.3.2 Monitoring OneWorld subsystems and jobs

Upon successful completion of the OneWorld startup sequence, the OneWorld subsystem should be active with jobs as indicated in the WRKACTJOB display (Figure 33).

```

Work with Active Jobs
ASM23
02/15/99 17:00:18
CPU %: .1 Elapsed time: 01:39:19 Active jobs: 197

Type options, press Enter.
 2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
 8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
   JDEB7332 QSYS SBS .0 DEQW
   JDENET_K ONEWORLD BCI .0 DEQW
   JDENET_K ONEWORLD BCI .0 DEQW
   NETWORK ONEWORLD BCH .0 PGM-JDENET_N SELW
   SENTINEL ONEWORLD ASJ .0 PGM-MONITOR SIGW
   QBATCH QSYS SBS .0 DEQW
   QCMN QSYS SBS .0 DEQW
   QCTL QSYS SBS .0 DEQW
   DSP01 JDE INT .0 PGM-J98INIT DSPW
More...

F11=Display elapsed data F12=Cancel F14=Include F24=More keys

```

Figure 33. Identifying OneWorld jobs with WRKACTJOB

This output shows that two kernel (JDENET_K) processes are running, along with the SENTINEL and NETWORK (JDENET_N) processes. The Server Administration Workbench (see 21.3.9.6, “Server Administration Workbench (SAW)” on page 655) provides additional detail on active OneWorld processes.

Batch requests appear in the QBATCH subsystem. For example, the WRKACTJOB SBS (QBATCH) command produces the results shown in Figure 34.

```

Work with Active Jobs
ASM23
02/15/99 17:00:18
CPU %: .1 Elapsed time: 01:39:19 Active jobs: 197

Type options, press Enter.
 2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
 8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
   QBATCH SBS .0 DEQW
   R0006P ONEWORLD BCH .0 DEQW

F11=Display elapsed data F12=Cancel F14=Include F24=More keys

```

Figure 34. Identifying OneWorld batch jobs

The R0006P job is the actual report running at this time. The program PRINTUBE is the job responsible for running and printing the request. When the job is finished, it leaves the queue, and the print job is either printed and deleted or saved in the B7332SYS/PRINTQUEUE file.

Part 2. Implementation

This part describes the implementation tasks and techniques necessary to install and properly set up OneWorld on the AS/400 system. With the exception of Chapter 17, “National language support” on page 471, all OneWorld-AS/400 customers will face topics described in this part. This part contains these chapters:

- Chapter 8, “Pre-installation and planning” on page 139, discusses general configuration and planning topics that you should consider when preparing to implement OneWorld on an AS/400 server.
- Chapter 9, “OneWorld installation and maintenance” on page 151, provides a sample step-by-step approach to a basic installation of OneWorld when using the AS/400 system as an Enterprise Server and the Integrated Netfinity Server as the Deployment Server. It also provides an overview of upgrade and update processes.
- Chapter 10, “WorldSoftware and OneWorld” on page 247, covers the coexistence and migration from WorldSoftware to OneWorld.
- Chapter 11, “Client Access Express and OneWorld” on page 283, offers a brief discussion of the benefits and features of Client Access Express. It also provides a brief overview of the differences between Client Access Express and Client Access/400.
- Chapter 12, “Operations” on page 299, discusses OneWorld operations on the AS/400 system. Administrators interested in understanding the AS/400 management of day-to-day operations and commands used to perform maintenance will benefit the most from this chapter.
- Chapter 13, “OneWorld package management” on page 327, describes the processes used to maintain, promote, and deploy custom modifications created with OneWorld Tools.
- Chapter 14, “Printing” on page 379, contains a general overview of AS/400 native printing as well as printing setup and issues specific to OneWorld.
- Chapter 15, “Security” on page 415, describes AS/400 security concepts and offers recommendations on how to setup security in OneWorld environment.
- Chapter 16, “Backup and recovery” on page 455, covers the basic concepts of backup and recovery on an AS/400 system and some strategies specific to OneWorld.
- Chapter 17, “National language support” on page 471, provides information on the national language support topics you need to understand when using OneWorld on an AS/400 system. This chapter is most beneficial for those readers who are implementing OneWorld on AS/400 in a multilingual or non-English speaking environment.

Chapter 8. Pre-installation and planning

This chapter discusses planning and pre-installation topics and activities that should be performed before you install OneWorld on an AS/400 server. It also covers prerequisites, such as correct hardware, software, infrastructure, and so on. As with any software installation process, pre-installation planning should be taken very seriously. A well prepared plan will save you considerable time and effort in the later stages of the implementation.

It is likely that many of the planning tasks mentioned in this chapter will remain relevant to future releases of OneWorld. However, more specific information contained in the *OneWorld Installation Guide (AS/400 Systems)*, such as required operating system releases and disk capacities, will almost certainly change from release to release. Note that the information regarding required PTFs and Service Packs is regularly updated as it becomes available. Therefore, before commencing the planning of your OneWorld installation, it is critical that you obtain the latest hardware and software requirements information from the J.D. Edwards Knowledge Garden at: <https://knowledge.jdedwards.com>

After entering your user and password information, use the menu structure on the left-hand side to navigate to the Product subdirectory. After the subdirectories are displayed, select **OneWorld Online** and then **Minimum Technical Requirements**.

You should also refer to the J.D. Edwards Worldwide Advanced Technologies Web site, which can be accessed through the Services tab on the Knowledge Garden Home Page. This site includes breaking news about new functionality, as well as links to useful tips and techniques relevant to OneWorld installation and configuration.

Note

These are secured Web sites only available to J.D. Edwards Business Partners. If you do not have access to these sites, contact J.D. Edwards Worldwide Customer Support.

Further information regarding OneWorld known issues, tips and techniques and Software Action Requests (SARs) that may affect your installation processes can be found on the Web at: <http://www.jdedwards.com/solutions/index.asp>

Access to this Web site is available to all J.D. Edwards customers.

IBM supplies Informational Authorized Problem Analysis Reports (APARs) containing details of all PTFs required to facilitate OneWorld support on the AS/400 server. Refer to 22.2.1, "Informational APARs" on page 717, for details on Informational APARs.

These APARs are updated regularly and, therefore, should be referenced periodically both during installation planning and post-installation.

8.1 OneWorld customer preparation

J.D. Edwards has identified several infrastructure planning and preparation actions that must be performed by the customer prior to installing OneWorld. These actions include checks to ensure that all hardware and software required to support the Configurable Network Computing (CNC) architecture are in place. A customer checklist is provided in the *OneWorld Installation Guide (AS/400 Systems)* that details these actions.

Note

J.D. Edwards deems that it is the responsibility of the customer to ensure that the hardware and software meet OneWorld's minimum requirements and to perform any preparation tasks that may be necessary, such as operating system upgrades and pre-requisite PTF installations.

8.2 Hardware requirements

This section briefly details the Deployment Server, Enterprise Server, and client workstation hardware required to support the OneWorld application at Release B7.3.3. More detailed information is given in the *OneWorld Installation Guide (AS/400 Systems)*.

The precise hardware requirements for your installation depend largely on the type of network environment in place and the configuration options you select. For example, if you choose to adopt the three-tiered approach, with logic processing performed on a separate system from the database, you may want to consider using the IBM OptiConnect product to increase performance. In a multi-site network where OneWorld data and processing will be distributed across a WAN, you may consider installing Workgroup Servers at each of the remote locations. By setting up data replication between the Enterprise Server and the Workgroup Servers and storing some static data closer to the client systems, network traffic can be reduced, thereby improving response times. There are many possible configuration choices and, therefore, thorough research must be performed before selecting which is most suitable for your installation.

For information regarding the configuration options recommended by J.D. Edwards, refer to Chapter 5, "OneWorld implementation designs" on page 71.

When planning any OneWorld implementation system, sizing is of fundamental importance. You must be sure that your system configuration can cope with both day-to-day processing requirements and occasional periods of extremely high processing, such as at financial year-end. A common mistake made by World customers moving to OneWorld is to assume that the system resource requirements are the same. The two products use the AS/400 system in different ways, with OneWorld typically demanding significantly more system resources than World. IBM and J.D. Edwards have established the International Competency Center (ICC) at J.D. Edwards' headquarters in Denver, Colorado, which provides worldwide sales and sizing support. Tools have been developed at the center for performing system sizing estimates for two-tier OneWorld implementations.

IBM and J.D. Edwards provide a sizing questionnaire that should be completed and submitted to have an estimate of your AS/400 server hardware requirements.

8.2.1 Deployment Server

The only processor architecture currently supported by OneWorld to act as a Deployment Server is Intel running Windows NT. The minimum hardware requirements of this server are:

- Intel Pentium 166 Mhz processor
- 128 MB RAM
- Ethernet or Token-Ring LAN connection
- CD-ROM drive
- Monitor with 800 x 600 pixel screen resolution
- Printer attached

You must also ensure that your Deployment Server hardware supports Windows NT Server 4.0. To verify this, check Microsoft's Hardware Compatibility List (HCL) for Windows NT. This can be found on the Web at:

<http://www.microsoft.com/isapi/hwtest/hcl.idc>

This site contains a detailed list of information regarding tested and approved hardware components, such as disks, network adapters, and so on.

Note that this server can be either a stand-alone Windows NT server, such as a Netfinity server, or an Integrated Netfinity Server (formerly the Integrated PC Server) installed on the AS/400 system.

Note

J.D. Edwards recommends that you ensure the integrity and performance of the Deployment Server and that it is dedicated to the deployment and maintenance of OneWorld software.

8.2.2 AS/400 Enterprise Server

J.D. Edwards' OneWorld application runs on the PowerPC AS/400 system. Both traditional AS/400 systems models and server models support this architecture. However, OneWorld's client/server design means that the server models usually deliver the best performance. After IBM introduced the new AS/400e server models in February 1999, there was no distinction between the advanced server models and advanced system models. The new AS/400e 7xx servers are ideal for both interactive and client/server applications.

The AS/400 Enterprise Server must include:

- Ethernet or Token-Ring LAN adapter
- Printer connection (IP addressable PostScript, line, or PCL)

These requirements also apply to any other AS/400 server that you may want to include in your CNC configuration, such as a Workgroup Server or a separate logic or Database Server.

8.2.3 Client workstations

Client workstation minimum hardware requirements depend on the function of the system, that is whether the system will be used by production users only or for the development of customized OneWorld programs. These requirements are:

- Intel Pentium 166 MHz processor
- 64 MB RAM (96 MB of RAM is recommended)
- 1.34 GB hard disk space (for a full package load without development objects)
- 1.65 GB hard disk space (for a developer workstation with a full package load)
- Printer connection (PostScript, line, or PCL)

8.2.4 Windows Terminal Server

Microsoft Windows NT Terminal Server Edition allows multiple remote users to connect to a server which appears to the Enterprise Server as one client system. In a WAN environment, this reduces the amount of network traffic generated between the Enterprise Server and the remote workstations.

The Windows Terminal Server (WTS) has the following minimum hardware requirements:

- Dual Pentium Pro 200
- 500 MB RAM
- 12 GB hard disk

This configuration is recommended to provide support for a maximum of eight client systems. If a higher number of users must be supported, add an additional 40 MB of RAM per user. Each user consumes a minimal amount of disk space on the WTS. Around 100 MB per user should be assumed for planning purposes.

J.D. Edwards states that they expect an implementation of 20 remote OneWorld users connected to a 4-way, 1 GB RAM WTS to experience a performance equivalent to that provided within a normal OneWorld CNC LAN environment.

8.2.5 Java Application Server

Java Application Server allows OneWorld to be used over the Internet. The only processor architecture that can currently support this functionality is Intel running Windows NT.

8.3 Software requirements

This section details the Deployment Server, Enterprise Server, and client workstation software required to support the OneWorld application at Release B73.3.

8.3.1 Deployment Server

The minimum software requirements for Deployment Server are:

- Microsoft Windows NT Server 4.0 with Service Pack 5 (use the 128-bit version with the Integrated Netfinity Server card if possible).
- Microsoft Access 97
- Microsoft Visual C++ 6.0 with Service Pack 1

- DB2 UDB for AS/400, Microsoft SQL Server 7.0, or Oracle 8.0.5
- Client Access Express

Note

If you are using MS SQL Server or Oracle, we recommend that you have a suitably qualified database administrator install and prepare the chosen Deployment Server RDBMS and be present onsite during the OneWorld software installation. If you are using DB2 UDB for AS/400, which is a part of OS/400, this is not necessary.

8.3.2 AS/400 Enterprise Server

For OneWorld Version B733.2, OS/400 V4R1M0 or a later version must be installed on the AS/400 server, along with the following licensed programs:

- 5769TC1: TCP/IP Connectivity Utilities for AS/400
- 5769CX2: ILE/C Compiler for AS/400
- 5769XE1: Client Access Express for Windows

We recommend that you always use the latest version of the OS/400.

Among all Client Access Express features, only the ODBC driver is mandatory for OneWorld support. This is currently supplied free of charge. Therefore, no licenses are required for Client Access if you are using no other parts of the product.

All OneWorld objects can be stored on the AS/400 server. The DB2 UDB for AS/400 database, supplied as an integral part of the operating system, is used for storing most OneWorld objects, including the central objects. OneWorld stream files are stored in the AS/400 Integrated File System (IFS) directories. Check the J.D. Edwards Web site for OneWorld minimum technical requirements.

J.D. Edwards recommends that the latest cumulative PTF package is always loaded on the AS/400 server before commencing a OneWorld installation. This must be installed along with the latest version of the database service pack SF9910 n (where n represents the release of OS/400 V4 installed on the system, for example SF99103 for OS/400 V4R3M0) and SF99104 for OS/400 V4R4M0.

There are also several individual PTFs required for each release of OS/400 to facilitate OneWorld support on the AS/400 server. These are listed in the *OneWorld Installation Guide (AS/400 Systems)* for the currently supported releases of OS/400. However, J.D. Edwards recommends that you reference their Web site and the IBM Informational APARs for more recent information.

8.3.3 Client workstations

Client workstation software requirements are:

- Windows 95, Windows 98, or Windows NT Workstation 4.0 with Service Pack 5
- Client Access V3R1M2 or greater with the latest Service Pack
- Microsoft MDAC 2.1 (required for SQL Server connectivity only)

- Microsoft Visual C++ 5.0 with Service Pack 1 (required for developer workstations only)
- SQL Server 6.5 or 7.0 client components or Oracle 7.3 or 8.0 client components

Note: Up-to-date Client Access Service Pack information can be found on the Web at: <http://www.as400.ibm.com/clientaccess>

Any client workstation to be used for connection to a OneWorld Java server must also include:

- Microsoft Internet Explorer 4.0 or greater
- Development Studio 5.0 with Service Pack 3 or greater Java Application Server

A server that is to run OneWorld Java Server must also include:

- Windows NT Server 4.0 with Service Pack 4
- Microsoft IIS 4.0 or greater
- JRun version 2.2

Note: JRun v2.2 can be downloaded in a ZIP file from Live Software's Web site at: <http://www.livesoftware.com/download>

8.4 Network assessment

The capabilities of your network infrastructure are of major importance to J.D. Edwards OneWorld due to the "network-centric" nature of the Configurable Network Computing (CNC) architecture upon which it is based.

OneWorld applications communicate across the various operating systems using the TCP/IP protocol. Therefore, your network hardware, such as routers, hubs, and interface adapters, must support this protocol.

When including Windows Terminal Servers in your configuration the WAN, use a minimum 56K line.

8.5 Disk space requirements

Calculation tables are provided in the *OneWorld Installation Guide (AS/400 Systems)* that indicate how much disk space is required on each hardware platform to support the OneWorld installation. The total amount of disk space required is site specific, depending on the number of path codes that are installed, the number of environments that are created, and how data is distributed within the CNC architecture. Therefore, it is critical that you perform accurate calculations before commencing the OneWorld installation to ensure that adequate disk space is available on all required platforms.

Note: OneWorld requires that all client workstations have a C: drive configured. This is referred to by several OneWorld kernel processes and cannot be altered. If a client system does not already have a C: drive, it must be created. Approximately 10 MB should be sufficient for OneWorld purposes.

8.6 Environment Checker

The Environment Checker is an application supplied by J.D. Edwards that performs several checks on the hardware and software environment, and reports any issues that may impact the installation of the OneWorld software. This application currently only supports Microsoft Windows/Intel platforms. Therefore, its use in an environment where the AS/400 server is to be used as the Enterprise Server is limited to Deployment Server verification.

8.7 Details worksheet

J.D. Edwards provides a worksheet in the *OneWorld Installation Guide (AS/400 Systems)* that should be used to record details of which OneWorld environments are to be created and where they are to be located. It is a comprehensive document that goes down to the data source and Object Configuration Manager (OCM) mapping level. Once completed, refer to this document during the Installation Planner phase of the installation.

Much of the information regarding data sources, environment, and OCM mappings is pre-defined on the worksheet. These contents assume that the J.D. Edwards recommended naming conventions will be used in your configuration. Adopting these naming conventions and pre-defined settings greatly simplifies the planning and installation process and should be used where possible.

8.8 Planning for coexistence with WorldSoftware

The major differentiating feature of OneWorld on an AS/400 server that separates it from other platforms is the ability for it to run in coexistence with J.D. Edwards WorldSoftware. A coexistent installation allows WorldSoftware and OneWorld to share the same database on the AS/400 server. Chapter 10, "WorldSoftware and OneWorld" on page 247, contains detailed information for running WorldSoftware and OneWorld in a coexistent environment.

From a configuration and planning perspective, there are a few considerations for WorldSoftware and OneWorld coexistence. The first consideration to be aware of is that there are dependencies between the WorldSoftware release and cumulative PTF level and the OneWorld release. For example, OneWorld Release B73.3 requires that the WorldSoftware be at Release A7.3 with cumulative PTF update 9 or greater. In addition to this, coexistence PTF update A73PC000X0 is required to enable support of the coexistent implementation.

8.8.1 Coexistent system data source considerations

In a coexistent configuration, the OneWorld system data sources contain tables that have the same name and function as WorldSoftware tables typically found in the security library, for example, the user and environment tables (F0092, F00921, F0093 and F0094). In addition, it contains other OneWorld specific libraries such as OCM mappings (F986101) and data source definitions (F98611). Although it is possible to share the common tables in a coexistent configuration, J.D. Edwards does not recommend this configuration. We recommend that you keep them separate by defining a System - B733 data source for all of your system tables that points to a new library (SYSB733).

8.8.2 Coexistence control tables

In a non-coexistent installation, all of your control tables are contained in a single data source. These tables are Next Numbers (F0002, F00021), User Defined Codes (F0004, F0004D, F0005, F0005D), and menus (F0082). In a coexistent environment, you typically share the Next Number and UDC tables between WorldSoftware and OneWorld. A typical coexistent plan configures the appropriate OCM mappings to point to the WorldSoftware AS/400 common library. When defining your plan, configure the AS/400 Common data source to point to your existing AS/400 Common library name. When the installation workbench is run, any OneWorld specific entries are added to these tables. Your existing WorldSoftware entries will be preserved.

8.8.3 Coexistence data dictionary

Even though the WorldSoftware and OneWorld data dictionary tables share the same name, they are in a different format. It is not possible to share a common data dictionary between WorldSoftware and OneWorld software. The OneWorld installation process installs a new pristine data dictionary in library DDB733. During the co-existent merge step of the workbench, a merge is run to bring any changes to the WorldSoftware data dictionary across to the OneWorld data dictionary. Ongoing synchronization of the two data dictionaries needs to be performed to keep the two data dictionaries in sync as changes are made to the WorldSoftware data dictionary.

J.D. Edwards supports the changing of WorldSoftware data dictionary items that are synchronized with the OneWorld data dictionary, but not the opposite. Consider this synchronization for your ongoing system administration planning, especially if you make a lot of changes to your data dictionary.

8.9 Central objects storage considerations

The DB2 UDB for AS/400 database, supplied as an integral part of the operating system, can be used for storing OneWorld objects. Prior to OS/400 V4R4, it was necessary to store Central objects on the Deployment Server in either Microsoft SQL Server or Oracle database. This restriction is removed at V4R3M0 and V4R4M0. Check the J.D. Edwards Web site for OneWorld minimum technical requirements.

8.10 Data source planning

The unique factor of data sources in an AS/400 OneWorld implementation is the need for “Do Not Translate” (DNT) data sources to access certain tables. Each AS/400 library that contains DNT tables must have a DNT data source that is configured not to perform translation. The following standard DNT data sources are created automatically during Installation Planner:

- System: B733 - DNT
- Business Data: *type* - DNT
- Versions: *pathcode* - DNT

The corresponding ODBC Data Sources, which you configure manually during Installation Workbench, as well as the *Enterprise Server - Server Map* data source, must be defined as “Do not translate CCSID 65535”.

8.11 AS/400 server preparation

There are several tasks that are necessary on the AS/400 system in preparation for your OneWorld installation. These tasks are explained in the following sections.

8.11.1 Configuring TCP/IP

Before the installation of OneWorld on the AS/400 server can commence, it is necessary to ensure that TCP/IP is configured, as OneWorld uses TCP/IP for all inter-system communications processing. For detailed information regarding the configuration of TCP/IP, refer to Appendix B, “TCP/IP basic installation and configuration” on page 781.

8.11.2 Verifying OS/400 security

OneWorld is currently incompatible with OS/400 security level 50. When this level of security is in place on the Enterprise Server (or any other AS/400 server which will process logic), OneWorld cannot successfully use certain APIs and, therefore, cannot function correctly. Use the following command to check the AS/400 system’s security level:

```
DSPSYSVAL QSECURITY
```

If the value is set at 50, it must be changed. However, before any change is performed, it must be established that it is acceptable for your security level to be reduced. This may necessitate a full audit of your AS/400 system security policy to ensure that the security provided by lower OS/400 security levels will be adequate.

8.11.3 Creating user ASP for OneWorld journals

To ensure the integrity and resilience of the AS/400 journals and journal receivers used by OneWorld’s Lock Manager application on the Enterprise Server, place them in a separate user auxiliary storage pool (ASP) from the OneWorld system and data libraries.

For instructions regarding how to create a user ASP, refer to Appendix A, “Adding a user auxiliary storage pool” on page 775.

8.12 OneWorld planning check list

The check list in Table 12 can be used as a basis for planning an initial OneWorld installation.

Table 12. Planning check list

Tasks to be completed	References/comments
Define OneWorld environments by resident platform-See the Customer Preparation chapter in <i>OneWorld Installation Guide (AS/400 Systems)</i>	See also “Typical Customer Configuration” in the <i>Configurable Network Computing Implementation Guide</i>
Pristine	
Conference Room Pilot	

Tasks to be completed	References/comments
Development	
Production	
Define OneWorld Data Sources	Illustrate with diagram
Distributed business logic	
Business data	
Control tables	
Data dictionary by release	
System Data	
Object Librarian by release	
Central Object specifications by release	
Server Map	
Versions	
As/400 data sources	
Define OneWorld applications/suites by resident platform	Illustrate with diagram
Document OneWorld Interfaces	
Other third-party products	
World co-existence	
Print distribution	
Document any non-OneWorld software that will share hardware	
Create inventory list	Include vendor contact data
Determine implementation schedule	
Define OneWorld application functions by resident platform	Illustrate with diagram
Deployment Server	
Enterprise Server	
Windows terminal server(s)	
Java Application Server	
Client workstations (heavy & lite clients)	
Define end user community requirements	
Service levels	
Develop end user census projections by application and server if possible.	Break down by application & server if possible
Determine response time expectations for each group	
Determine UBE (batch) throughput expectations	
Define periodic processes by application	Monthly, Quarterly, Fiscal Year End, etc.

Tasks to be completed	References/comments
Determine Availability expectations	
Define support expectations	
Determine cost of outages by department/functional area	
Determine Network Requirements	
Perform OneWorld Network Assessment	
Create graphical view of network topology to support OneWorld	
Create acquisition schedule of any additional required hardware/software	
Create implementation schedule of changes required to support OneWorld	
Determine Hardware Requirements	Using information from above tasks
Complete & Submit IBM/JDE Sizing & Planning Questionnaire	Obtain by dialing (800) IBM-4FAX or (408) 256-5422 if outside US or Canada and requesting Doc ID 8651. Complete and fax the questionnaire to (770) 659-5245 or e-mail it to ibmerp@us.ibm.com
Build hardware requirements inventory list	
Create acquisition schedule	
Determine data center physical requirements for hardware	
Create Physical Site Preparation Plan	
Create installation schedule	

Chapter 9. OneWorld installation and maintenance

This chapter provides a sample step-by-step approach for basic installation of OneWorld Release B733.2 when using:

- The AS/400 system with OS/400 V4R4 as an Enterprise Server
- The Integrated Netfinity Server with Windows NT as a Deployment Server

This chapter also includes an overview of the upgrade and cumulative update processes. The information provided in this chapter should be used in conjunction with the *OneWorld Installation Guide (AS/400 Systems)*, which is supplied with the OneWorld software. The *Installation Guide* is unique for each release of OneWorld due to the continuing evolution of the installation process. Where necessary, detailed supplemental information pertaining to the software installation is supplied in this chapter that is not currently available in the *OneWorld Installation Guide (AS/400 Systems)*.

OneWorld installation is a complex process, and as such, should be performed by a qualified OneWorld Installer by following the *OneWorld Installation Guide (AS/400 Systems)*.

This chapter was created from the perspective of giving a *general feel* for the type of tasks that you will perform and the type of problems you may encounter during the execution of your OneWorld installation. The majority of the tasks mentioned in this chapter are common to all types of OneWorld installations where an AS/400 serves as the Enterprise Server. The many hardware, software, and configuration options available when implementing OneWorld dictate that it is not possible to include all potential scenarios in this book. It is very likely that your particular installation will differ greatly from any other installation. The more you understand the configurable nature of OneWorld, the more sense you can make of the specific tasks required of you.

Following the completion of the installation processes, your Conference Room Pilot (CRP) phase may commence. It is during this phase that you test all aspects of your client/server environment, including performance (both server and network), security, and reliability. You may need to make changes to your configuration based on your findings during this testing. For example, it is possible that you may amend Object Configuration Manager (OCM) mappings within a WAN environment to allow certain static data tables to be located on a server closer to client systems remote from the Enterprise Server. The possibilities are numerous. The completion of your installation and initial configuration tasks is only the beginning of what is likely to be an in-depth implementation process.

9.1 First time installation

The information in this section is presented as a series of notes relating to each of the main installation processes. These main processes are defined in the *OneWorld Installation Guide (AS/400 Systems) Release B73.3.2* as:

- Installing the Deployment Server
- Planning the Installation
- Running Installation Workbench
- Installing the Enterprise Server

- Installing the Workstations
- Completing the Installation

The first step in the OneWorld installation process is to load the software onto the Deployment Server. From this central point, the software is distributed throughout the enterprise to all servers and clients. It is on this system that the configuration of OneWorld is planned, including the determination of what objects to install and where to install them. The software is then loaded onto all required servers through a combination of direct loads from CD-ROM, and file transfers from the Deployment Server. Deployment packages are created for each user to determine which OneWorld objects to install on the user's workstation. Finally, several post-installation tasks are performed to complete the OneWorld installation.

The following sections describe each of these processes in more detail and provide additional information where appropriate.

9.1.1 Installing the Deployment Server

The Deployment Server acts as the focal point for the OneWorld installation process and ongoing creation and deployment of OneWorld packages to client workstations. J.D. Edwards requires that the Deployment Server be an Intel based server running Windows NT. The Integrated Netfinity Server on the AS/400 meets these requirements and is certified by Microsoft to run Windows NT. The installation tasks described in this chapter are based on using the Integrated Netfinity Server on the AS/400 as the Deployment Server.

The software is loaded onto the Deployment Server by running the setup program on the first installation CD-ROM. The execution of the setup.exe program walks you through selecting the OneWorld components and Path Codes you are going to install. Path Code directory structures are created based on the installation options selected.

The Planner Path Code and environment are created on the Deployment Server for use during the Installation Planner and Installation Workbench phases. Together with the JDEPLAN.MDB Access database, the Planner environment serves as a "staging" area for the OneWorld installation process.

Starting and using the OneWorld Planner environment require that the software be authenticated with a security code. The Deployment Server installation process asks you to input an authorization code, which can be obtained from the J.D. Edwards Contracts Department by calling 1-800-289-2999 and then selecting option 5 for SPC code authorization. When asked for customer number, enter your customer number on your telephone keypad followed by the # key. Then select the option for OneWorld to get the Authorization code department.

Prior to V4R3M0 of OS/400, OneWorld Central Object tables had to reside in an Oracle or Microsoft SQL database, due to a restriction in the size of binary large object (BLOB) table fields that could be stored in the DB2 UDB for AS/400, the AS/400 integrated database.

Central Objects on the AS/400 system

OneWorld Central Objects no longer require an Oracle or Microsoft SQL database since they can be installed on the DB2 UDB for AS/400 with OS/400 V4R3M0 or later.

J.D. Edwards has recently created a workaround to overcome this restriction. This workaround involves dividing up BLOBs into 32 KB pieces and works with OS/400 V4R3 and later. Although OS/400 V4R4 can *natively* support larger BLOBs, J.D. Edwards is not expecting to take advantage of this native support until a future release of OneWorld.

Scripts are supplied with the OneWorld software that are used to create the databases, and the tables that are required on the Deployment Server. Once these are created, Central Object specifications and other (BLOBs) can be loaded into the databases.

Two Enterprise Server activities are also performed during the Deployment Server phase:

- The installation of the OneWorld Data Dictionary
- The configuration of TCP/IP that enables a communication between the Enterprise and the Deployment Server

9.1.1.1 Configuring the AS/400 Integrated Netfinity Server

Before starting the Deployment Server installation processes, the Windows NT Server must be configured. The following procedures describe how Windows NT is installed on the AS/400 Integrated Netfinity Server, so that it can be used as the OneWorld Deployment Server.

This section provides a basic quick start installation of the Integrated Netfinity Server. The installation consists of three steps:

1. Installing IBM licensed program AS/400 Integration Services for Windows NT
2. Installing Windows NT on the Integrated Netfinity Server
3. Completing Windows NT post installation activities

The process begins on the AS/400 system and completes on the Windows NT server. The AS/400 system controls the installation process and supplies default values for the NT installation portion. The redbook *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164, contains detailed installation and configuration information.

Note

The fast path installation described in this section is based on the Integrated Netfinity Server with OS/400 V4R4. J.D. Edwards recommends using only Windows NT Server Edition 4.0 with Service Pack 5 for the Deployment Server.

Installing AS/400 Integration Services for Windows NT

To use the Integrated Netfinity Server on the AS/400 system as the Deployment Server, AS/400 Integration Services for Windows NT must be installed.

To install AS/400 Integration Services for Windows NT on the AS/400 system, perform the following steps:

1. Sign on to the AS/400 system as QSECOFR (or any other user with *SECOFR authority).
2. Load the software installation CD containing the Integration Services for a Windows NT licensed program, and type GO LICPGM on an AS/400 command line. The Work with Licensed Programs screen is displayed as shown in Figure 35.

```
LICPGM                      Work with Licensed Programs                      System: SYSNAMXX
Select one of the following:

Manual Install
  1. Install all

Preparation
  5. Prepare for install

Licensed Programs
  10. Display installed licensed programs
  11. Install licensed programs
  12. Delete licensed programs
  13. Save licensed programs

                                                                    More...

Selection or command
====> 11

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=Information Assistant
F16=AS/400 Main menu
```

Figure 35. Work with Licensed Programs

3. Enter option 11 to obtain the display shown in Figure 36.

```

                                Install Licensed Programs
                                System:  SYSNAMXX

Type options, press Enter.
  1=Install

      Licensed  Product
Option Program  Option  Description

      5769SS1   26      OS/400 - DB2 Symmetric Multiprocessing
      5769SS1   27      OS/400 - DB2 Multisystem
      5769SS1   29      OS/400 - AS/400 Integration for NT
      5769SS1   30      OS/400 - QShell Interpreter
      5769SS1   31      OS/400 - Domain Name System
      5769SS1   32      OS/400 - Directory Services
      5769SS1   33      OS/400 - Private Address Space Environment
      5769SS1   34      OS/400 - Digital Certificate Manager
      5769SS1   35      OS/400 - Cryptographic Service Provider
      5769SS1   36      OS/400 - PSF/400 1-20 IPM Printer Support
      5769SS1   37      OS/400 - PSF/400 1-45 IPM Printer Support
      5769SS1   38      OS/400 - PSF/400 Any Speed Printer Support

More...

F3=Exit  F11=Display status/release  F12=Cancel  F19=Display trademarks

```

Figure 36. Install Licensed Programs

4. Locate the entry for the 5769SS1 licensed program, product option 29, and enter option 1 to perform the installation.
5. Upon completion of the licensed program installation, issue the Check Product Option (CHKPRDOPT) command to verify that the installation was successful:

```
CHKPRDOPT PRDID(5769SS1) OPTION(29)
```

The command returns messages detailing the status of the product installation.

Installation of the licensed program product (LPP) can also be verified with the DSPSFWRSC command (Figure 37 on page 156).

```

                                Display Software Resources
                                System: SYSNAMXX

Resource
  ID      Option  Feature  Description
5769SS1  25         5109    OS/400 - NetWare Enhanced Integration
5769SS1  25         2924    OS/400 - NetWare Enhanced Integration
5769SS1  26         5107    OS/400 - DB2 Symmetric Multiprocessing
5769SS1  27         5108    OS/400 - DB2 MultiSystem
5769SS1  29         5110    OS/400 - AS/400 Integration for NT
5769SS1  29         2924    OS/400 - AS/400 Integration for NT
5769SS1  30         5050    OS/400 - QShell Interpreter
5769SS1  30         2924    OS/400 - QShell Interpreter
5769SS1  31         5050    OS/400 - Domain Name System
5769SS1  31         2924    OS/400 - Domain Name System
5769SS1  32         5050    OS/400 - Directory Services
5769SS1  32         2924    OS/400 - Directory Services
5769SS1  33         5111    OS/400 - Private Address Space Environment
5769SS1  34         5050    OS/400 - Digital Certificate Manager

More...

Press Enter to continue.

F3=Exit   F11=Display libraries/releases  F12=Cancel
F19=Display trademarks

```

Figure 37. Display Software Resources (DSPSFWRSC)

Following the installation of AS/400 Integration Services for Windows NT, ensure that the latest cumulative PTF CD has been installed.

Installing Windows NT on the Integrated Netfinity Server

This section describes the fast path installation process for installing Windows NT on the Integrated Netfinity Server. The redbook *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164, contains detailed installation procedures.

Perform the following process:

1. Setup and verify the hardware installation.

Before beginning installation, connect the monitor, keyboard, and mouse to the appropriate ports on the Integrated Netfinity Server card.

2. Verify that the AS/400 Integration with Windows NT Server software is installed.

Use the Work with Licensed Programs menu (you can access it by typing the `GO LICPGM` command) to verify whether the AS/400 Integration with Windows NT Server is installed. Option 10 displays the installed licensed programs, and option 11 installs selected options from the AS/400 CD-ROM drive.

3. Turn on automatic performance adjustment.

Do this to allow the AS/400 system to automatically adjust the machine pool size to accommodate the Integrated Netfinity Server. Type:

```
CHGSYSVAL SYSVAL(QPFRADJ) VALUE(3)
```

4. Make the CD-ROM available.

Insert the Windows NT Server 4.0 CD into the AS/400 CD-ROM drive and ensure that the AS/400 CD-ROM is varied on. Type:

```
WRKCFGSTS *CTL OPT01
```


5. Install the Windows NT server (`INSWNTSVR`).

This is a long running command that boots and re-boots the server card in the process. This command causes a virtual Token-Ring address to be automatically assigned, which is required for completing the Windows NT configuration.

While the command is running, a number of informational messages are displayed including messages similar to the following examples:

- `INSWNTSVR` is a long running command. Please wait.
- 160M of 200M bytes initialized for disk image `AS2NWS`.
- 288M of 1000M bytes initialized for disk image `AS2NWS`.
- Operation in progress, performing hardware reset.
- Specify 192.168.002.002 as the IP Address for the Virtual Token-Ring Adapter.

6. Insert the Windows NT Server Edition 4.0 CD in the AS/400 CD-ROM drive.

7. Type `INSWNTSVR` on a command line, and press F4.

This starts the installation process. The command, as specified in Figure 38, and Figure 39 and Figure 40 on page 158, installs a Windows NT server on the Integrated Netfinity Server with the name `AS2NWSA` and IP address 172.20.10.247. It will be a member server in the workgroup `JDEGRP`.

```
Install Windows NT Server (INSWNTSVR)

Network server description . . . NWSD          > AS2NWS
Resource name . . . . . RSRNAME             > CC08
Domain role . . . . . DMNROLE              > *SERVER
Windows NT version . . . . . WNIVER        *NT40
Install option . . . . . OPTION            *INSTALL
Port 1:
  Line type . . . . .                       *NONE
  Local adapter address . . . . .
  Maximum transmission unit . . . . .      1492
  AS/400 internet address . . . . .        *NONE
  AS/400 subnet mask . . . . .             *NONE
  NT internet address . . . . .           > '172.20.10.247'
  NT subnet mask . . . . .                 > '255.255.255.0'
```

Figure 38. Install Windows NT Server (Part 1 of 3)

```

Install Windows NT Server (INSWNTSVR)

Port 2:
Line type . . . . . PORT2 *NONE
Local adapter address . . . . .
Maximum transmission unit . . . . . 1492
AS/400 internet address . . . . . *NONE
AS/400 subnet mask . . . . . *NONE
NT internet address . . . . .
NT subnet mask . . . . .
TCP/IP local domain name . . . . . TCPDMNNAME *SYS
TCP/IP name server system . . . . . TCPNAMSVR *SYS
+ for more values
Server message queue . . . . . MSGQ > AS2NWS
Library . . . . . > QGPL

```

Figure 39. Install Windows NT Server (Part 2 of 3)

```

Install Windows NT Server (INSWNTSVR)

Event log . . . . . EVTLOG *ALL
+ for more values
Server storage space sizes: SVRSTGSIZE
Drive D size . . . . . 200
Drive E size . . . . . > 1000
Convert to NTFS . . . . . CVNTFS > *YES
To workgroup . . . . . TOWRKGRP > JDEGRP
To domain . . . . . TODMN
Full Name . . . . . FULNAM > IBM
Organization . . . . . ORG > IBM
Language version . . . . . LNGVER *PRIMARY
Synchronize date and time . . . . . SYNCTIME *YES
Windows NT license key . . . . . WNTLICKEY 111-1111111
License mode: LICMODE
License type . . . . . > *PERSERVER
Client licenses . . . . . > 25
More...

```

Figure 40. Install Windows NT Server (Part 3 of 3)

A brief discussion of the changes and considerations for certain parameters follows. If no changes are indicated for a parameter, use the default value.

- **Resource name (RSRCNAME):** Use `DSPHDWRSC *CMN` to determine the correct Integrated Netfinity Server hardware resource name.
- **Networking information (PORT1):** Contact your network administrator for IP address and subnet mask information. The line type is set based on the type of network (`*TRN16M`, `*TRN4M`, `*ETH100M`, `*ETH10M`).
- **NT message queue (MSGQ, EVTLOG):** Specify a message queue to provide more detailed messages of the activities regarding the Integrated Netfinity Server. Using `*NONE` causes only serious messages to be sent to QSYSOPR that will be interspersed with all of the other activities of the AS/400 system. Using the job log creates spooled files that are sometimes difficult to locate, and show the activities for the current session only.

Use the Event Log parameter (EVTLOG) to specify which Windows NT server event log messages (system, application, security, or all) are sent to the AS/400 message queue.

- **NT domains and workgroups (DMNROLE, TOWRKGRP, TODMN):** Due to the work that it will carry out, the Deployment Server should not be set up as a Primary or Backup Domain Controller. Set the domain role to *SERVER.

If you want to propagate AS/400 users and groups to the Windows NT server or domain, you need to give the domain administrator rights to the User Administration Service on the Windows NT server on the Integrated Netfinity Server. As an alternative, you can install an additional Integrated Netfinity Server to function as the primary domain controller.

For small networks and in test environments, a workgroup installation may be sufficient. Read the *Windows NT Server* manual for more information on setting up and planning for NT domains.

- **Server storage space (SVRSTGFSIZE, CVTNFS):** The INSWNTSVR command creates a *server storage space*. Drives C:, D:, and E: are added to this storage space:
 - The C: drive contains Windows NT boot code plus a bootable DOS image.
 - The D: drive contains copies of the installation code (the \i386 folder) and programs, files, and device drivers specific to AS/400 Integration with Windows NT server and 200 MB of storage should be sufficient.
 - The E: drive is the Windows NT system drive and assigning 1 GB of storage provides space for additional Windows software. If more Windows software programs than discussed here will be added later to this disk drive, this server storage space may need to be even larger. Ensure that you assign sufficient storage at this time to accommodate these programs.
 - The C: and D: drive are created as FAT partitions, and the *Convert to NTFS* option gives you the option of converting drive E: to NTFS. Due to security and performance considerations, the E: drive should be converted to NTFS. Additional user storage spaces should also be formatted as NTFS. The only caveat is that if there is a problem with Windows NT, it will not be possible to recover files by booting the C: drive.
- **Synchronize date and time (SYNCTIME):** Specify *YES to synchronize the time of the Windows NT Server on an Integrated PC Server with the time of the AS/400 system during a vary on and every 30 minutes. A value of *NO synchronizes the time only during a vary on.

On the AS/400 system, be sure that system value QUTCOFFSET has the same value as Windows NT. On Windows NT, the Automatically adjust for daylight savings changes option is not checked, because the AS/400 system does not automatically adjust for daylight savings time and we do not want the AS/400 and Windows NT server to become unsynchronized.

- **Licensing information (WNTLICKEY, LICMODE):** Enter your Windows NT license key and license mode based on your license agreement.

Job log

The INSWNTSVR command creates a job log that contains information about any problems encountered during the installation. The job log should be saved. The IP address assigned to the Windows NT side of the virtual LAN is contained in this job log.

8. Complete the installation of Windows NT.

After the AS/400 INSWNTSVR command is complete, the Integrated Netfinity Server is automatically varied on. TCP/IP on the AS/400 system is started (if it was not already), and Windows NT on the Integrated Netfinity Server starts. The Windows NT signon appears on the monitor attached to the Integrated Netfinity Server. Sign on Windows NT on the Integrated Netfinity Server as Administrator, and complete the following steps:

- a. Select **I Agree** when the Software License Agreement is displayed.
- b. Complete the Name and Organization fields when prompted, if it was not entered on the INSWNTSVR command above.
- c. Establish an administrative password, if required, and enter it when prompted for Administrator Account.
- d. Select **OK** in response to either or both of the messages:
 - Error - Unattended Setup: The IP Address key in the TCP/IP section of the database must be set to a value.
 - Error - Unattended Setup: The subnet mask that you have entered for the IP address is not valid.
- e. Configure Network Setup:
 - i. When the Microsoft TCP/IP Properties prompt is displayed, you need to enter your assigned IP Address, Subnet Mask, and Default Gateway address for your Ethernet (or Token Ring) adapter that will communicate with your LAN/WAN (we used an Ethernet adapter as shown in Figure 41). If necessary, contact your network administrator for your network configuration information.
 - ii. Select the **DNS** tab and make the server a member of the appropriate workgroup (or domain) for your network.
 - iii. Select the **WINS Address** tab, if necessary for your network, to enter the Primary and Secondary WINS Server addresses.
 - iv. Deselect **Enable LMHOSTS Lookup**.
 - v. Before selecting OK, return to the **IP Address** tab. Select adapter **[2] AS/400 Virtual Token Ring Adapter**.
 - vi. Enter the IP Address which the AS/400 system assigned for the virtual internal LAN (usually in the form 192.x.x.x) during the running of the INSWNTSVR command. Enter the subnet mask 255.255.255.0.
 - vii. Leave the Default Gateway blank.
 - viii. When you are sure all of your entries are correct, select **OK**.

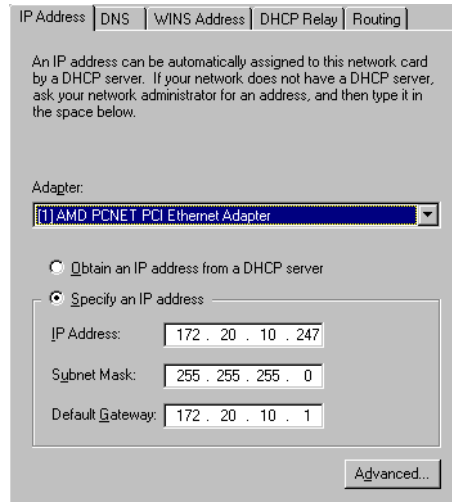


Figure 41. Microsoft TCP/IP Properties

- f. If you see a message warning you that one of the adapter cards has an empty primary WINS address, ignore it. Select **Yes** to continue.
- g. When the Date/Time Properties form is displayed, select the **Date & Time** tab to verify the settings and correct if necessary.
- h. Select the **Time Zone** tab to select the correct time zone for your installation.
- i. Deselect the box corresponding to **Automatically adjust clock for daylight saving changes**, because the AS/400 system does not automatically adjust for daylight savings time and you want to ensure that the time on your servers stays synchronized.
- j. Setup the correct display adapter. If you receive a prompt that the system detected a video adapter in your machine, select **OK**.
- k. Select the correct display settings. Select **Test**.
- l. Select **OK** when prompted that the new mode will be tested.
- m. Select **Yes** if the test bitmap displayed correctly.
- n. Select **OK** to save the settings that you just tested.
- o. Select **OK** when prompted for the Display Properties.
A number of operations are then performed automatically before your Windows NT machine re-boots.
- p. When Windows NT finishes re-booting, sign on as *Administrator*.
- q. Right-click **Network Neighborhood**. Select **Properties**.
- r. If you are using an Ethernet adapter on your Integrated Netfinity Server, select the **Adapters** tab.
- s. Select the **AMD Ethernet Adapter**.
- t. Select **Properties**.
- u. For the M11 Mode, select **100Mb F.D.**. This changes the adapter to 100 MB per second and full duplex, which has been tested to be faster than the default.

- v. Select **OK** to confirm the change.
- w. If desired, select the **Protocols** tab. Remove any unneeded protocols (only TCP/IP and AS/400 HostLan Bridge Driver 1 protocols are required).
- x. After saving your settings, select **Yes** to the prompt to restart your computer.

Windows NT automatically re-boots.

Windows NT post installation activities

There are a number of steps that you need to perform after you install, configure, and connect the AS/400 system with the Windows NT server. Follow these steps:

1. Install the latest service pack (at the time of this writing, this was Service Pack 5) on the Windows NT server:
 - a. After inserting the Service Pack 5 CD in the AS/400 CD-ROM drive, use Windows NT Explorer to find the CD.
 - b. Double-click the file **MSNT128-sp5** to run the service installation program.
 - c. Select **Setup** on the Welcome display.
 - d. Select **I Accept** when the Software License Agreement is displayed.
 - e. Click **Install**.
 - f. Click **Restart**, when prompted with the message Windows NT Service Pack 5 Installation is complete.
 - g. Select **Yes** on the Shutdown confirmation display.

The Windows NT server re-boots.

2. When the Windows NT server finishes re-booting, perform a Service Pack Level Check on the Windows NT server. To perform this function, click **Start->Programs->AS/400 Windows NT->Level Check**. This ensures that you are using the latest Service Pack.
3. Verify successful communications. From an MS-DOS command screen, ping the adapter card, the AS/400 system, and other systems on the network using PING or TRACERT followed by the IP address or system name.
4. Install Client Access Express. For OneWorld on the Deployment Server to communicate successfully with the AS/400 Host Server, the Client Access ODBC driver must be installed on the Deployment Server.

Note

Use the Client Access *Express* ODBC and *not* the ODBC drivers from previous Client Access clients such as Client Access Windows 95/NT. Previous ODBC drivers are not supported with OneWorld.

- a. Load the Client Access Express CD in the AS/400 CD-ROM drive.
- b. Using Windows NT Explorer, find the CD drive, find the **Setup.exe** file. Double-click **Setup.exe** to run.
- c. Click **Next** on the Client Access Welcome display.
- d. Select **Yes** to agree to the conditions on the Client Access Express License Information display.

- e. For Type of Installation, select **Typical** if Data Transfer and 5250 Display and Printer Emulation are required (both of these options require a Client Access License for each concurrent display on which they are used). Otherwise, select **Custom** and select only those Client Access Express components required (at least Client Access ODBC *must* be selected).
- f. Select the default for Destination Folder (or select **Browse** to choose an alternate directory). Select **Next**.
- g. Select **Next** for Programs Menu Shortcut.
- h. Select **Next** for Copying Files.
- i. Deselect the **README** file. Select **Next** on Install Completed display.
- j. Click **Finish** on Setup Complete display to restart your computer now.
- k. Click **Yes** on Shutdown Confirmation display.

Windows NT re-boots.

5. Install Latest Client Access Service Pack. Obtain the latest Client Access Express service pack from the Client Access Web site <http://www.as400.ibm.com/clientaccess> and place on a shared network drive or order the service pack using the Send PTF Order (`SNDPTFORD`) command.
 - a. Using Windows NT Explorer, locate the Client Access Express service pack **Setup.exe** file (either on CD or a shared network drive). Double-click the **Setup.exe** file to run it.
 - b. Deselect **View the README file**. Select **Next** on the Welcome display.
 - c. Click **Next** on the Copying Files display.
 - d. Click **Finish** on Setup Complete display.
6. Allocate AS/400 system disk storage for use by Deployment Server.

Use the Create Network Storage (`CRTNWSSTG`) command to allocate sufficient network storage space, as required, for the Windows NT server. The maximum size for each Network Storage Space is 8 GB. However, multiple network storage spaces can be defined and combined into a single volume set. For more information on combining network storage spaces into a single volume set, refer to step 8 on page 165.

Be aware that the `CRTNWSSTG` command is a long running command.

We are creating one 2 GB network storage space for the Windows NT temporary work space, since this is approximately double the 1 GB of RAM on our Integrated Netfinity Server. Depending on the amount of RAM on your Integrated Netfinity Server, ensure that you create a network storage space approximately double this size.

We also recommend that at least two (more may be required) 8 GB network storage spaces be created for storing the required OneWorld environments on the Windows NT Deployment Server. Review your J.D. Edwards OneWorld sizing for your disk space requirements on the Deployment Server. The disk space requirements for loading OneWorld on the Deployment Server *must* be doubled to provide sufficient disk resources for future OneWorld release upgrades. Additional disk space is also required for the deployment of packages.

Repeat this process for each network storage space to be created:

a. Run the AS/400 command:

```
CRINWSSTG NWSSTG(storage space) NWSSIZE(mb) FORMAT(*NTFS)
TEXT('description')
```

Here, *storage space* is the name of the network server storage space to be assigned to the Windows NT server, and *mb* is the number of MB required for this network storage space.

This command can be submitted to batch for multi-threading, where multiple network storage spaces need to be created, to free up the interactive session, but may not reduce the total time required for creating them.

b. While the CRTNWSSTG command is running, OS/400 generates a number of messages. These messages are displayed on the terminal from which this command is run, if run interactively, or to the job log associated with this job, if submitted to batch:

- Create of NWS storage space in progress. Please wait.
- Network server storage space created.

```
                                Create NWS Storage Space (CRINWSSTG)

Type choices, press Enter.

Network server storage space . . > AS2NWS1      Name
Size . . . . . > 2000                          1-8000
Format . . . . . > *NTFS                        *HPFS, *FAT, *NETWARE...
Text 'description' . . . . . > 'Network Storage Space 1'
```

Figure 42. Create NWS Storage Space

7. Assign drive space to the network server card.

OneWorld requires that the OneWorld software on the Deployment Server fit into one continuous storage space. You have to repeat this procedure to allocate enough additional storage spaces to Windows NT to meet OneWorld's disk space requirements. Once the storage spaces are created, a Windows NT volume set combining individual storage spaces has to be formatted.

After allocating the space for Windows NT, you must link it to the Network Server. To link the disk space, you must first vary off the Windows NT server.

- Locate the Windows NT server by issuing the `WRKCFGSTS *NWS` command.
- Select option **2** to vary off the appropriate Windows NT server.
- A message may appear in the System Operator queue. If it does, respond with a `G`.
- Use the `ADDNWSSTGL` command (press F4 for prompting) to link the storage space and network server, for each of the network storage spaces created. Alternatively, you can use the `WRKNWSSTG` command and enter option `10` to link the storage spaces to the network server. You then see a display similar to the example in Figure 43. Enter the desired name for Network server storage space (usually assign a name that can be incremented for each network storage space). Enter your assigned name of the server (which you assigned in step 5 on page 157) for Network server description.

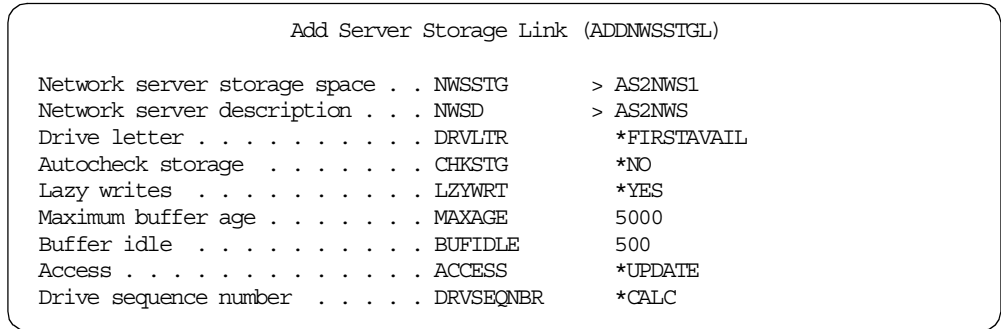


Figure 43. Add Server Storage Link

8. Format allocated disk space to Windows NT.

Once you allocate the space in OS/400, you can now format this space in Windows NT. To format the storage spaces for use with Windows NT server, follow these steps:

- a. Go to the Windows NT Disk Administrator. Click **Start->Programs->Administrative Tools->Disk Administrator**.
- b. You receive a message similar to the example in Figure 44. Click **OK** to continue.

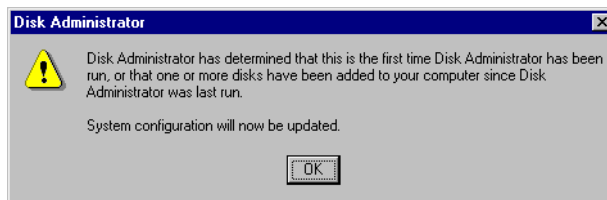


Figure 44. Windows NT Disk Administrator warning

- c. For each network storage space linked to the Integrated Netfinity Server, you receive a message similar to the example in Figure 45. Click **Yes** for each message.

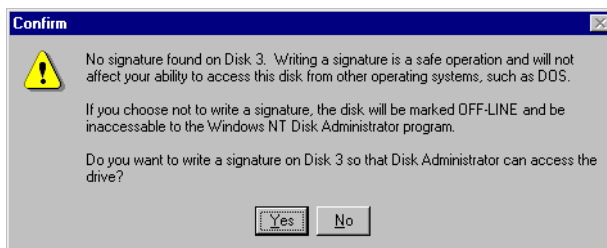


Figure 45. Windows NT disk signature confirmation warning

- d. Each disk drive which shows as Unknown needs to be formatted to be usable by Windows NT. Repeat for each disk drive requiring formatting. For each of these disk drives, perform the following steps:
 - i. Right-click the disk drive, and select **Format**.
 - ii. On the Format display, enter `NTFS` for the File system, select **Quick Format** (Figure 46 on page 166). Click **Start** to format the disk.

- iii. Click **OK** on the warning since there is no data on the disk drive to erase.
- iv. Click **OK** when formatting is complete.

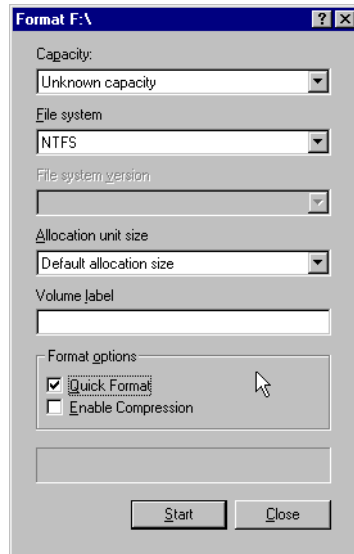


Figure 46. Windows NT Disk Format Prompt

- e. To combine the 8 GB disks into one contiguous disk space, delete the existing disk space first by right-clicking on each newly allocated disk and selecting **Delete**.
- f. Click **Yes** in reply to the warning message, since there is no data on these disk drives to be lost.

Note: Do not delete the C:, D:, or E: drives!

After performing this procedure, the disk space will show as *free space*.

- g. To make one contiguous volume of 16 GB, or more, by combining multiple disk storage spaces, hold down the Ctrl key and click each disk you want in the volume set.
- h. After all disk spaces are selected, right-click and select **Create Volume Set**.
- i. On the Create Volume Set display, enter the maximum total size shown in the Create Volume Set of total size field. Click **OK**.
- j. Right-click again, and select **Commit Changes Now**.
- k. Click **Yes** to save the changes made to the disk configuration.
- l. Click **Yes** to restart the computer to make the changes take effect.
- m. Click **OK** to the Disks were update successfully message.
- n. Click **OK** to initiate system shutdown.
- o. Click **Yes** to Shutdown Confirmation.

The Windows NT server re-boots.

- p. Once the server is back up, sign on as *Administrator*. Go back into the Disk Administrator, and format the volume set drive as NTFS, as displayed in Figure 46.

9. Install Microsoft Data Access (MDAC).

Microsoft Data Access is required for OneWorld installation on the Deployment Server. It is a part of Microsoft Office 97, and it can also be downloaded from Microsoft Web site. We installed Microsoft Office, which includes Microsoft Access 97, as follows:

- a. Load the Microsoft Office 97 CD in the AS/400 CD-ROM drive.
- b. If the setup does not automatically start, use Windows NT Explorer to find the **Setup.exe** command. Double-click **Setup.exe** to run it.
- c. Click **Install Microsoft Office**.
- d. Click **Continue** on the Welcome to the Microsoft Office 97 installation program display.
- e. Enter the appropriate name and organization on the Name and Organization Information display. Select **OK**.
- f. Click **OK** the Confirm Name and Organization Information display.
- g. Click **OK** the Microsoft Office 97 Setup Product ID display.
- h. On the next Microsoft Office 97 Setup display, change the folder where Microsoft Office 97 is to be installed, if necessary. Click **OK**.
- i. Select **Typical** for type of installation on the next Microsoft Office 97 Setup display.
- j. Click **Continue** on Microsoft Office 97 - Typical display.
- k. Click **OK** when Microsoft Office 97 Setup was completed successfully is displayed.

10. Install the C++ Compiler.

OneWorld requires the installation of C++ compiler on the Deployment Server. We installed Visual Studio, which includes C++.

- a. Load the Visual Studio (which includes the C++ compiler) CD in the AS/400 CD-ROM drive.
- b. Using Windows NT Explorer, find the CD drive. Find the **Setup.exe** command in the *Vc32.cd* folder. Double-click **Setup.exe** to run it.
- c. Click **Next** on Installation Wizard for Visual C++ 6.0 Professional Edition display.
- d. Select **I accept the agreement** on the End User License Agreement display. Click **Next**.
- e. Enter the appropriate name and company name on the Product Number and User ID display. Click **Next**.
- f. If an out of date version of Internet Explorer is detected, click **Next** on the Install Internet Explorer 4.01 display.
- g. Click **Next** on the Internet Explorer 4.01 SP1 Active Setup display.
- h. Select **I accept the agreement** on the License Agreement display. Click **Next**.
- i. Keep the defaults on the Installation Option display. Click **Next**.
- j. Keep the defaults on the Windows Desktop Update display. Click **Next**.

- k. Select appropriate country and language on the Active Channel Selection display. Click **Next**.
 - l. If desired, change the folder where Internet Explorer 4.01 will be installed on the Destination Folder display. Click **Next**.
 - m. Click **No** to overwrite the newer file prompt on the Confirm File Replace display (occurs twice).
 - n. Click **OK** on the Setup has finished installing components display.
 - o. Click **OK** in response to the message *Setup must restart your computer.* Windows NT re-boots.
 - p. Log on Windows NT as *Administrator*. The Windows Internet Explorer setup automatically completes.
 - q. Click **Next** on the Installation Wizard for Visual C++ 6.0 Professional Edition display.
 - r. Enter the folder name in which to install the C++ compiler on the Choose Common Install Folder display. Click **Next**.
 - s. Click **Continue** on the Visual C++ 6.0 Professional Setup Welcome display.
 - t. Click **OK** on the Visual C++ 6.0 Professional Setup Product ID display.
 - u. Select **Typical** for the installation type on the next Visual C++ 6.0 Professional Setup display.
 - v. On the Setup Environment Variables display, click **OK** without selecting *Register Environment Variables*.
 - w. Click **OK** on the Windows NT Debug Symbols display.
 - x. Click **Restart Windows** on the Visual C++ 6.0 Professional - Restart Windows display to complete the installation of C++.
Windows NT re-boots.
 - y. Log on Windows NT as *Administrator* and allow the installation of Visual C++ to automatically complete.
 - z. On the Install MSDN (Microsoft Developer Network Library) display, deselect **Install MSDN**. Click **Next**.
 - aa. Click **Next** on the Other Client Tools display.
 - ab. Click **Finish** on the Register Over the Web Now! display.
11. Install the Visual Studio 6.0 Service Pack 3.
- a. Load the Visual Studio C++ Service Pack CD in the AS/400 CD-ROM drive.
 - b. Using Windows NT Explorer, find the CD drive and the **Mdac_typ.exe** file. Double-click **Mdac_typ.exe** to install MDAC 2.1 SP1.
 - c. Click **Yes** on the Microsoft Data Access 2.1 License Agreement display.
 - d. Click **Continue** on the Microsoft Data Access 2.1 Setup Welcome display.
 - e. Click **Complete Install all Data Access Components** on the next Microsoft Data Access 2.1 Setup display to start the installation.
 - f. Click **Restart Windows** on the Microsoft Data Access 2.1 - Restart Windows display to complete the installation.
Windows NT re-boots.

- g. Logon as NT Administrator. Using Windows NT Explorer, find the **Setup3.exe** file. Double-click **Setup3.exe** to start the installation of the service pack.
- h. Click **Continue** on the Visual Studio 6.0 Service Pack 3 Setup Welcome display.
- i. Click **I Agree** on the License Agreement display.
- j. Click **OK** on the Visual Studio 6.0 Service Pack Setup display.
- k. Click **Restart Windows** on the Visual Studio 6.0 C++ - Restart Windows display to complete the installation.
Windows NT re-boots.
- l. Logon as the Windows NT Administrator.

9.1.1.2 Running Environment Checker

Environment Checker is a standalone application that you run before installing OneWorld. Environment Checker diagnoses any OneWorld configuration or setup issues that you may have at the operating system level. For example, it verifies that you have enough disk space to install OneWorld to various machines.

When you run Environment Checker, the interface prompts you for information about your system. Once the application finishes, Environment Checker creates an output report, which contains information about your system and any warning messages that need attention.

The current Version 1.2 of Environment Checker supports Microsoft Windows NT and Intel platforms only so can only be used for checking the Deployment Server and Client Workstations.

Here are the steps for running the Environment Checker:

1. Insert the OneWorld Setup CD into the AS/400 CD-ROM drive.
2. On the Deployment Server, using Windows NT Explorer, click the CD drive. Open the CD Templates folder, and click **Environment Checker**. Double-click **ec_winintel.exe** to run the Environment Checker program.
3. Type the directory where you want the results created. Press the Enter key.
4. Type **N** in reply to the prompt for using the previous set of answers. Press the Enter key.
5. Type **B73.3** (not B73.3.2) in reply to the prompt for the OneWorld version being used. Press the Enter key.
6. Type **1** to indicate that this is the Deployment Server. Press the Enter key.
7. Type **3** to indicate that the Deployment Server will run in an AS/400 environment. Press the Enter key.
8. Type **D** in reply to the prompt for which database product will be used on the Deployment Server. Press the Enter key. Even though no database product will be used on the Deployment Server, a valid answer is required.
9. Type the number of path codes (that is the number of OneWorld environments) that you will be loading. Press the Enter key.
10. Type the name of the AS/400 Enterprise Server. Press the Enter key.

11. Type 3 in reply to the prompt to get the most detail on the Environment Checker report. Press the Enter key.
12. The Environment Checker runs. Press the Enter key to view the ec_rpt.txt file for warnings.
13. Ensure that you resolve all warnings before proceeding with the OneWorld Deployment Server install. Note that since you are using this Deployment Server in an AS/400 environment, you can ignore Warning 0009 indicating that the required ANSI C Compiler, Microsoft C++ 6.0 is not installed.

9.1.1.3 Client Access TCP/IP configuration tasks

On the AS/400 system, use the `CFGTCP` command, and enter option 1 (Work with TCP/IP interfaces) to identify both the virtual *TRLAN and the physical LAN IP addresses. You use these IP addresses to define the Client Access sessions. The Integrated Netfinity Server Client Access needs to be configured using the IP address assigned to the virtual *TRLAN. For all other client machines, use the actual IP address of the external LAN.

Setting up Client Access Express to use the internal LAN

You can set up Client Access Express for Windows to use the internal LAN. This minimizes the risk of connection loss due to an external network failure. If you are using a low speed LAN or have heavy network traffic on your LAN, it may also improve the performance of ODBC communications between Windows NT on the Integrated Netfinity Server and the AS/400 system. To do so, perform these steps:

1. Find the AS/400 system's internal LAN TCP/IP interface. Type `NETSTAT *IFC` on an AS/400 command line, and press Enter. Search for an entry with a line description with the same name as your network server description but ending in 00. The TCP/IP address of that entry should have the form 192.168.x.y. Write down this address.
2. Create a new Client Access Express for Windows connection. Follow these steps:
 - a. Use Operations Navigator to create a new connection.
 - b. Right-click the new connection and select **Properties**.
 - c. Click the **Connection** tab.
 - d. In the IP address lookup frequency window, select **Never - Specify IP address**.
 - e. Type the IP address of the internal network address in the IP Address window.
 - f. Click **OK**.

After you create that connection, you can use the other Client Access Express for Windows/AS/400 functions in the same way as on any other PC.

9.1.1.4 Setting the Deployment Server paging file size

Before starting the Deployment Server installation processes, the Windows NT Server paging file size should be set to optimize system performance. To achieve this, perform these steps:

1. Select **Start->Settings->Control Panel->System**.
2. Select the **Performance** tab.

3. Click on the **Change** button in the Virtual Memory section.
4. Select the disk drive that will be used for the Paging File. Use the disk drive assigned to the network storage created in step 6 on page 163.
5. Set the Initial Size for the Paging File Size field to a number equal to the amount of RAM available on the system.
6. Set the Maximum Size to around double the Initial Size. Take care not to set the maximum number too high on systems where there is limited overall disk space, since this will have a negative affect on system performance. The display will appear similar to the example in Figure 47.
7. Click **Set** to complete the settings.
8. Click **OK** to save the settings. Click **Yes** in reply to the prompt to restart the computer since Windows NT must re-boot for the new settings to be used.
9. Click **Yes** in reply to the Shutdown Confirmation message.
Windows NT re-boots.

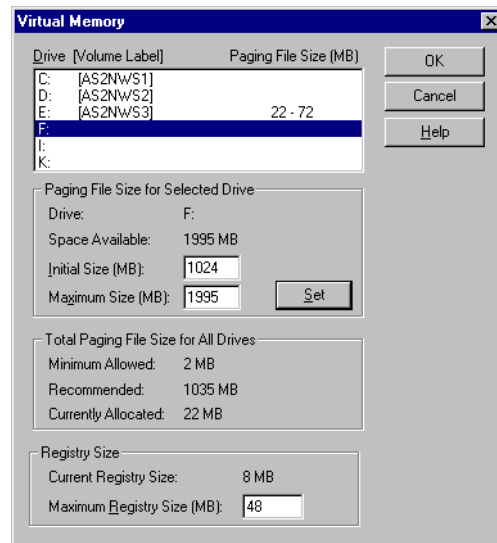


Figure 47. Setting Paging File Size on the Deployment Server

9.1.1.5 Configuring the OneWorld Deployment Server

Before you install OneWorld, you must configure the Deployment Server. To do so, complete these steps:

1. Create the user JDE as a system administrator on the Deployment Server. Sign on the Deployment Server as *Administrator*.
 - a. Select **Start->Programs->Administrative Tools->User Manager for Domains**.
 - b. Select **User->New User** to be able to enter a new user. The display looks similar to the example in Figure 48 on page 172. Enter the user *JDE* with the password *JDE*.
Note that passwords are case sensitive.
 - c. Deselect **User Must Change Password at Next Logon**.

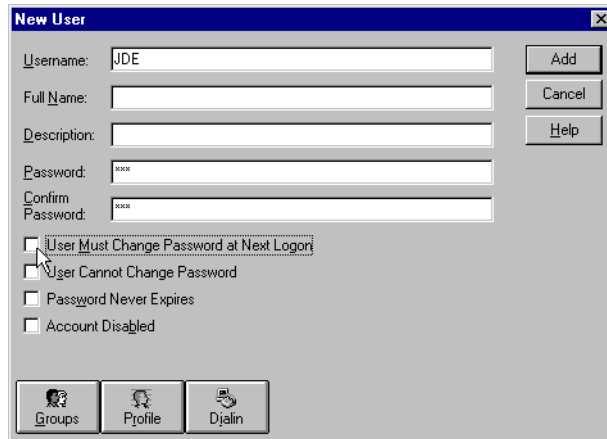


Figure 48. Creating a new Windows NT user

- d. Click **Groups** to make this new JDE user a member of the Administrator group.
 - e. Click **Administrators** to highlight.
 - f. Click **Add** to make JDE a member of the Users and Administrators groups.
 - g. Click **OK** to accept this change to the groups and return to the New User display.
 - h. Click **Add** to confirm the new user JDE.
 - i. Select **Start->Log Off Administrator**.
 - j. Select **Yes** to Log Off Windows as Administrator.
 - k. Log on again as user JDE.
2. Create a printer device for use by the Deployment Server.
 - a. Right-click the **Network Neighborhood** icon on the desktop.
 - b. Double-click **Entire Network**.
 - c. Double-click **Microsoft Windows Network**.
 - d. Double-click the workgroup of which the Deployment Server was made a member with the INSWNTSVR command.
 - e. Double-click the print server.
 - f. Enter an appropriate user ID (for example, Administrator) and password to connect to the print server. Select **OK**.
 - g. Double-click the printer which should be added for the Deployment Server.
 - h. If prompted for Windows to set up the printer, select **OK**. Windows adds this printer to the selectable printers for the Deployment Server.

As an alternative method for creating the printer device, perform the following steps:

 - a. Select **Start->Settings->Printers**.
 - b. Double-click **Add Printer**.
 - c. Select **Network Printer Server**.

- d. If necessary, expand the workgroup of which the Deployment Server was made a member with the INSWNTSVR command in step 5 on page 157.
 - e. Click the printer to highlight it, and click **OK**.
 - f. Select **Yes** or **No** based on whether this printer should be used as the default Windows printer, and click **Next**.
 - g. Click **Finish** in reply to the message that the network printer has been successfully installed.
3. Install file transfer protocol (FTP) on the Deployment Server.
 - a. Select **Start->Settings->Control Panel**.
 - b. Double-click **Network**.
 - c. Select the **Services** tab.
 - d. If Microsoft Internet Information Server 3.0 is not listed in the Network Services, click **Add**.
 - e. Double-click **Microsoft Internet Information Server**.
 4. Configure FTP.
 - a. Using Windows NT Explorer, find the disk drive where Windows NT resides (probably E:).
 - b. Find the **inetmgr.exe** file in the \WINNT\system32\inetrv directory path. Double-click **inetmgr.exe** to run it.
 - c. Right-click the name of the Deployment Server corresponding to the FTP Service.
 - d. Select the **Service** tab.
 - e. On the display similar to the example in Figure 49, select **Allow Anonymous Connection**. Deselect **Allow only anonymous connections**. Click **OK**.

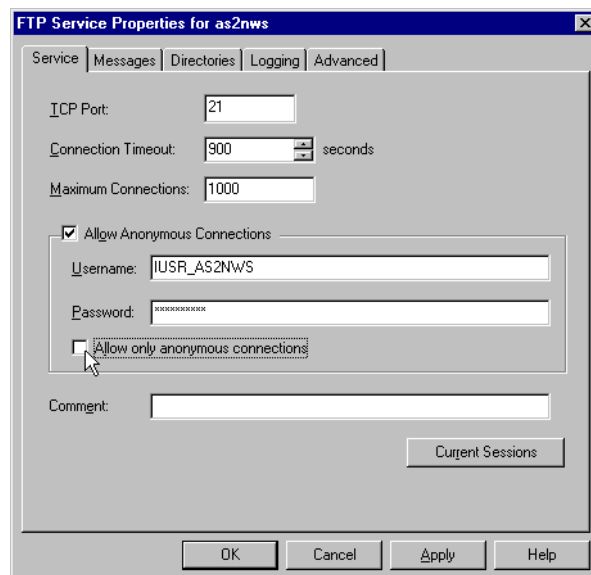


Figure 49. Configuring FTP Service Properties on the Deployment Server

- f. Click **Yes** in reply to the Internet Service Manager warning.

- g. Click the **Directories** tab.
- h. If the drive on which OneWorld is going to be installed does not appear, click **Add** to see a display similar to the example in Figure 50.

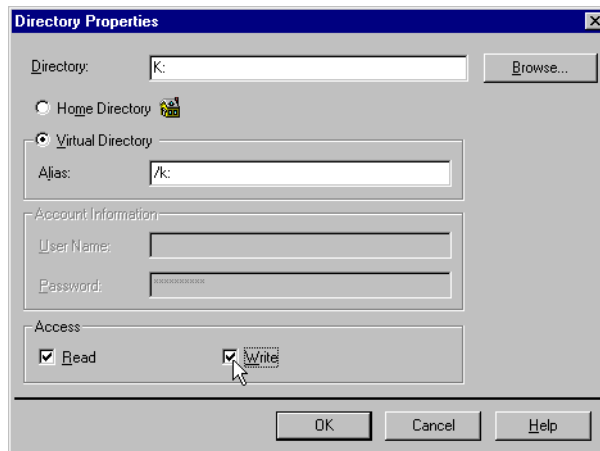


Figure 50. Configuring FTP Directory Properties on the Deployment Server

- i. Enter the directory (for example `K:`), where OneWorld is going to be installed on the Deployment Server.
- j. Enter the same disk drive preceded by a slash for Alias (for example, `\k:`). Select **Read** and **Write** Access.
- k. Click **OK**.
- l. Click **OK** on the FTP Services Properties display to accept the new FTP configuration.

9.1.1.6 Installing platform-independent objects

The first OneWorld installation CD is used at this point to load the platform independent objects onto the Deployment Server. Upon completion of the load, the installer is given the option to install Adobe Acrobat Reader V4.0 and Microsoft Internet Explorer V4.01, which are also supplied on the CD. These are both required on the Deployment Server to enable certain aspects of OneWorld functionality.

Upon completion of the setup program, verify the OneWorld directory structure. The expected structure is listed in Chapter 3 of the *OneWorld Installation Reference Guide (All Systems)*. This includes directory paths for your selected Path Codes.

The second installation CD contains the setup scripts for each RDBMS compatible with OneWorld. These are used later in the Deployment Server installation process to create the databases necessary to contain the Central Object tables and other BLOBs tables. The installer is prompted to select the chosen RDBMS, either Microsoft SQL Server or Oracle Workgroup Server. Since we will install the Central Objects tables in DB2 UDB for AS/400, we do not use these scripts here.

1. Sign on the Deployment Server as `JDE`.
2. Insert the first OneWorld setup CD into the AS/400 CD-ROM drive. Setup starts automatically.

3. Click **B7332 Deployment Server Install**.
4. Click **Next** on the OneWorld installation program Welcome.
5. When the Welcome to J.D. Edwards OneWorld Setup! display appears, record the serial number. Contact the J.D. Edwards Contracts Department (call 1-800-289-2999, extension 4535) and provide the serial number and your customer number. You are provided an Expiration Date, Number of Licenses, and Authorization Code. Enter these in the corresponding fields, and click **Next**. If a new display appears, the correct authorization code has been entered.

Attention

If you cancel the OneWorld Installation program before it completes successfully, you need to obtain a new authorization code.

6. You see a display similar to the example in Figure 51, if Adobe Acrobat Reader version 4.0 or Microsoft Internet Explorer version 4.01 are not installed. Select each application listed, and click **Next**. Click **Next** on the Acrobat Reader 4.0 Setup Welcome display. Click **Accept** on the Software License Agreement display. If necessary, browse to find a suitable directory on the Choose Destination Location. Click **Next**. Click **OK** on the Thank you for choosing Acrobat Reader display.

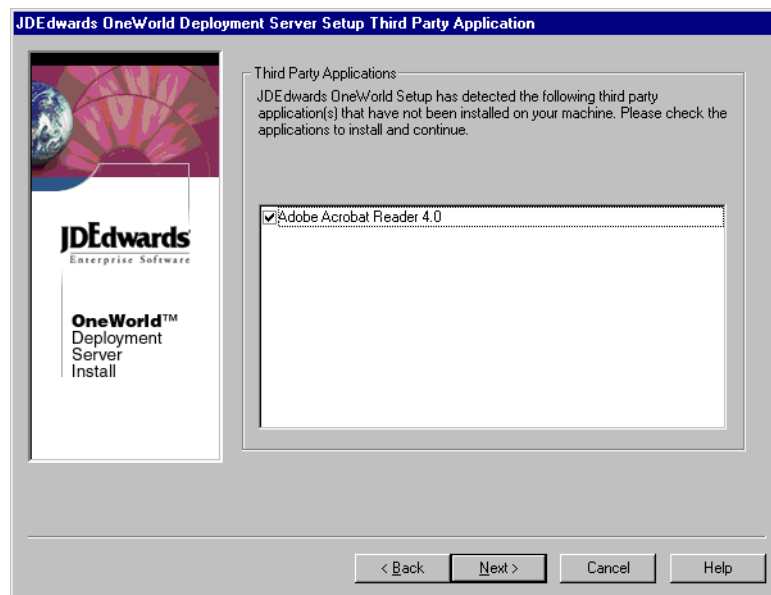


Figure 51. OneWorld Deployment Server Adobe Acrobat Reader Installation

7. Select **Custom** for the J.D. Edwards OneWorld Deployment Server Setup Type (refer to Figure 52 on page 176). This option is required to install the Central Objects on the AS/400 system. Enter the directory where the OneWorld files will be installed on the Deployment Server (for example, K:\JDEdwardsOneWorld\B733). If this directory does not already exist, the setup program creates it. Click **Next**.

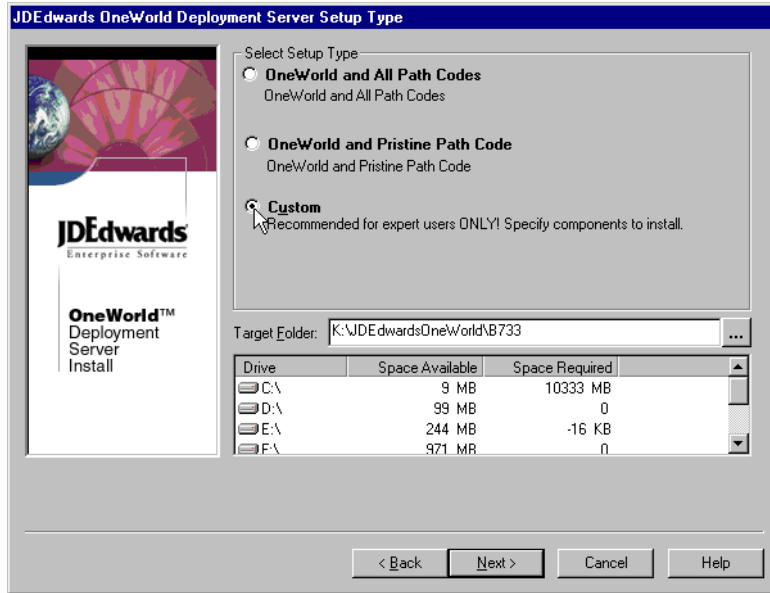


Figure 52. OneWorld Deployment Server Setup custom installation

8. The setup program verifies that there is sufficient disk space for installing OneWorld on the disk drive requested. If there is sufficient disk space, click **Next** to continue. Otherwise, choose another disk drive, click **Next**, and repeat this step.
9. Select the path codes corresponding to the OneWorld environments that you plan to install (refer to Figure 53). If all path codes are automatically selected, deselect those that you do not want. Click **Finish**.

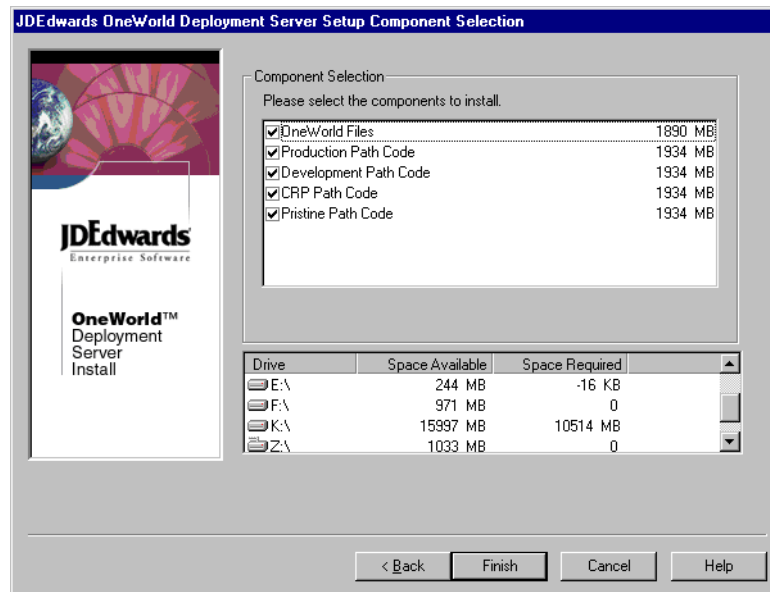


Figure 53. OneWorld Deployment Server Setup Component Selection

10. Click **OK** when you see the message stating that the J.D. Edwards OneWorld Deployment Server Setup succeeded.

11. Cancel out of the OneWorld Installation Manager, because the installation of the first setup CD is finished.
12. Insert the second setup CD into the AS/400 CD-ROM drive.
13. When the J.D. Edwards OneWorld Installation Manager display appears, click **B7332 Deployment Server Install** to begin the installation of the second CD.
14. Click **Next** on the J.D. Edwards OneWorld Deployment Server Setup Welcome display.
15. Select **Custom** on the J.D. Edwards OneWorld Deployment Server Setup Type display (Figure 52), because the Central Objects will be installed on the AS/400 system. Click **Next** to continue.
16. Deselect both **SQL Server** and **Oracle** on the J.D. Edwards OneWorld Deployment Server Setup Component Selection display (Figure 54), because the Central Objects are being installed on the AS/400 system. Click **Finish**.

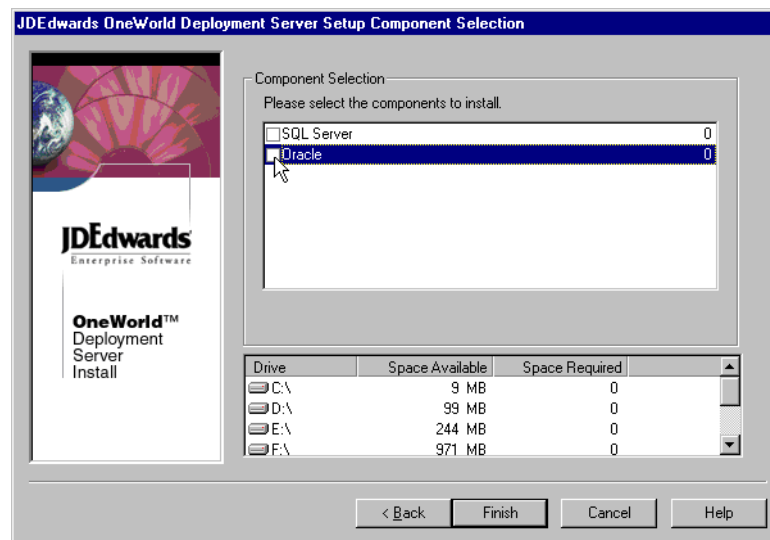


Figure 54. OneWorld Deployment Server Setup Central Objects Component Selection

17. Click **OK** when you see the message stating that the J.D. Edwards OneWorld Deployment Server Setup succeeded.
18. Cancel out of the OneWorld Installation Manager. The installation of the second setup CD is now finished.

9.1.1.7 Sharing the directory \JDEdwardsOneWorld B73.3

Using Windows NT Explorer, find the \B733 directory in the J.D. Edwards OneWorld directory on the Deployment Server disk drive where you installed OneWorld. Then, complete the following steps:

1. Right-click the **\B733** directory.
2. Select **Sharing**.
3. Select **Shared As**.
4. Type **B733** as the Share Name. Your display should look like the example in Figure 55 on page 178.
5. Click **OK**.

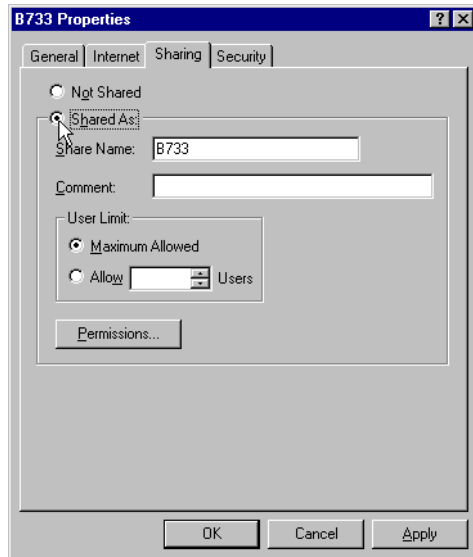


Figure 55. Setting up disk share for \\JDEdwardsOneWorld\B733 directory

Note

The *OneWorld Installation Guide (AS/400 System)* suggests setting up Windows NT security on the Deployment Server OneWorld directories at this time. This should *not* be done. The security of the Deployment Server OneWorld directories should only be completed after the successful installation of the Enterprise Server and at least two client workstations. This makes it easier to determine the cause if there are any problems setting up further client workstations following the security setup.

9.1.1.8 Installing libraries and Data Dictionary on the AS/400 system

At this stage, the OneWorld Data Dictionary must be created on the AS/400 Enterprise Server. At Release B73.3 and later, the Data Dictionary library must *always* be loaded from the installation CD-ROMs. At previous releases of OneWorld, this was necessary for coexistence installations only.

This process also creates all AS/400 libraries required by each of the One World environments supplied by J.D. Edwards. Verify that the following libraries are created:

- B733MAP
- DDB733: Contains Data Dictionary tables for OneWorld
- PathCodeB733
- PathCodeB733DNT: Contains the versions tables for the OneWorld path code
- PathCodeCTL: Contains menu tables for OneWorld
- PathCodeDTA: Contains data tables for OneWorld, and WorldSoftware in a co-existence environment, with the exception of PRISTDTA. PRISTDTA only contains data tables for OneWorld, as WorldSoftware does not have a Pristine environment.
- JDEOW

- OLB733
- SYSB733

Here, *PathCode* represents the PathCode identifier (CRP for Conference Room Pilot, DEV for development, PRIST or PRS for Pristine, PROD or PRD for production, and TEST for test) for each of the OneWorld environments that you are installing. There is *no* DEVDTA library, as the development and test environments share the same data. There is *no* PRISTCTL library, as the menu tables for the Pristine environment are found in the JDEDTA library. There is no DEVCTL library since the development and test environments share the same menu tables. There is no TESTB733DNT since the test and CRP environments share the same versions tables.

To install the AS/400 libraries and the Data Dictionary, complete the following steps:

1. Sign on to the AS/400 system as security officer (user ID QSECOFR).
2. Insert the AS/400 Direct Enterprise Server Install 1 of 4 CD into the AS/400 CD-ROM drive.
3. Enter the LODRUN OPTxx command on the AS/400 system, where *OPTxx* is the name of the CD-ROM drive. This creates all of the libraries listed above, except the Data Dictionary. Then the menu A98OWMNU (Figure 56) appears on the AS/400 display.

```

A98OWMNU                J.D. Edwards & Company                DENAS2
  B733                  OneWorld Installation Menu

... DATA LIBRARIES
2.  Install Data Dictionary
3.  Install Central Objects

... SYSTEM LIBRARIES
4.  Install OneWorld Server Libraries
5.  Install Additional Path Codes

... PRINTER CONFIGURATION
7.  Create Output Queue
8.  Create Job Queue

... POST INSTALLATION
9.  Edit INI File
10. Verify Library List

... SERVER CONFIGURATION
12. Porttest
13. Start JDE Server
14. End JDE Server
15. Clear IPC

... MISC. OPERATIONS
17. Display Active Jobs
18. Work with Spooled Files

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
F13=Information Assistant  F16=AS/400 main menu

```

Figure 56. AS/400 OneWorld Installation Menu

4. Enter option 2, and press Enter to install the Data Dictionary. The AS/400 display appears like the example in Figure 57 on page 180.

```

Install Data Dictionary (C980WDD)

Type choices, press Enter.

Device Name . . . . . OPT01      Character value
Base Release Level . . . . . B733      B733
Double Byte . . . . . *NO          *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 57. Install OneWorld Data Dictionary on the AS/400 system

5. If necessary, change the Device name to that of your AS/400 CD-ROM drive (this is usually OPT01). Press Enter.

Note

AS/400 libraries may be created for environments that have not been selected for installation. They are not automatically deleted. If you do not want them on your AS/400 system, you must delete them manually (use `DLTLIB` command).

9.1.1.9 Setting up the TCP/IP protocol for the Enterprise Server

TCP/IP should have already been configured on the AS/400 system during the preparation of the infrastructure. It is necessary at this stage of the installation process to perform further configuration tasks to allow the Deployment Server to communicate with the AS/400 Enterprise Server.

1. Sign on to the AS/400 system as security officer (`QSECOFR`).
2. Enter `CFGTCP` and press Enter to access the Configure TCP/IP menu.
3. Choose option 10 and press Enter to Work With TCP/IP Host Table Entries.

For more information on TCP/IP configuration on AS/400 system, refer to Appendix B, “TCP/IP basic installation and configuration” on page 781.

Creating an Enterprise Server host table entry

The AS/400 Enterprise Server must have an entry in its TCP/IP host table in the format `hostname.domain name`. To create this entry, follow these steps:

1. Find the IP address for the Enterprise Server. If it exists, choose option 5 and press Enter, to verify that the format is correct, as shown in Figure 58.


```

                                Display TCP/IP Host Table Entries
                                System: ASM23
Internet Address . . . . . : 5.1.72.146
Host names:
  Name . . . . . : ASM23

  Name . . . . . : ASM23.RCH.IBM.COM

Text description . . . . . :

```

Figure 58. Display Host Table Entries display

2. If the entry does not exist, enter option 1 and press Enter to create one.
3. If there is an entry, but it is in another format, an alias can be added to it by choosing option 2 and pressing Enter to add the correct format.

Note: The TCP/IP host name for the Enterprise Server is the same as the current system name. Use the Display Network Attributes (DSPNETA) command to verify this, as shown in Figure 59.

```

                                Display Network Attributes
                                System:  ASM23
Current system name . . . . . : ASM23
  Pending system name . . . . . :
Local network ID . . . . . : ITSC
Local control point name . . . . . : ASM23
Default local location . . . . . : ASM23
Default mode . . . . . : BLANK
APPN node type . . . . . : *NETNODE
Data compression . . . . . : *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : *LCLNETID  *ANY

                                More...

Press Enter to continue.

F3=Exit  F12=Cancel

```

Figure 59. Display Network Attributes

Creating a Deployment Server host table entry

A host table entry must also be created for the Deployment Server on the AS/400 Enterprise Server. To create this entry, follow these steps:

1. Enter option 1 and press Enter to add a new host table entry for the Deployment Server.

2. Type the IP address of the Deployment Server (the same address that was specified on the INSWNTSVR command in step 5 on page 157) in the Internet Address field. Enter the name of the Deployment Server (the name that was specified on the INSWNTSVR command in step 5 on page 157) in the Host name field. Press Enter.

After adding this entry, issue a Verify TCP/IP Connection (`PING`) command from the AS/400 server using the system name, not the TCP/IP address, of the Deployment Server. This ensures that successful communications can be established between the two systems.

Deploying server FTP configuration verification

The File Transfer Protocol (FTP) is used during the Installation Workbench and host load processes to transfer the OneWorld System data source and software from the Deployment Server to the AS/400 Enterprise Server. To ensure that FTP is configured correctly on the Deployment Server, an FTP session should be established from the AS/400 server to the Deployment Server, as well as from the Deployment Server to the AS/400 server, using the `JDE` account to signon.

Perform the following steps on the AS/400 system to test FTP functionality:

1. Type `FTP NTserver` on an AS/400 command line, where `NTserver` is the name of your Deployment Server on the Integrated Netfinity Server. Press Enter to start FTP.
2. Type `JDE` in reply to the FTP user ID prompt. Press Enter.
3. Type the password for the JDE user ID on the Deployment Server. Press Enter.
4. If the connection is successful, type `QUIT`. Press Enter to end the FTP connection.

Perform the following steps on the Windows NT server on the Integrated Netfinity Server to test FTP functionality:

1. Type `FTP servername` in an MS-DOS session, where `servername` is the name of your AS/400 server. Press Enter to start FTP.
2. Type `JDE` in reply to the FTP user ID prompt. Press Enter.
3. Type the password for the JDE user ID on the Deployment Server. Press Enter.
4. If the connection is successful, type `QUIT`. Press Enter to end the FTP connection.

These commands are also issued by the system during the host load process. Verifying their success now should avoid potential connectivity problems later.

9.1.1.10 Copying your specifications and installation plan files

To protect your Deployment Server installation in case of failure, J.D. Edwards recommends that you copy your `JDEKRNL.XDB` and `JDEKRNL.DDB` specifications files and your `JDEPLAN.MDB` installation plan file to a temporary directory. If the original files are corrupted later, you will have backup copies of these files.

1. Create a temporary directory for these backups if you do not already have one.

- Using Windows NT Explorer, copy the JDEKRNL.XDB and JDEKRNL.DDB files from the *n*:\JDEwardsOneWorld\B733\System\Bin32 directory to the temporary directory, where *n* is the drive location of the OneWorld software on the Deployment Server.
- Using Windows NT Explorer, copy the JDEPLAN.MDB file from the JDEPLAN.MDB file from the *n*:\JDEwardsOneWorld\B733\Planner\Data directory to the temporary directory, where *n* is the drive location of the OneWorld software on the Deployment Server.

9.1.1.11 Configuring Microsoft Access ODBC

J.D. Edwards recommends that the buffer size for the Microsoft Access ODBC driver on the Deployment Server is set at 2048 Kb per processor on the Deployment Server. The default value for this is 1024 Kb.

- To set this allocation, click **Start->Settings->Control Panel->ODBC Data Sources->User DSN**.
- If you do not see the User Data Source MS Access 97 Database, click **Add**. Select **Microsoft Access Drive (*.mdb)**. Click **Finish**.
- Double-click **MS Access 97 Database**. From the ODBC Microsoft Access Setup display, select **Options**. Update the **Buffer Size** field as displayed in Figure 60. Click **OK**.

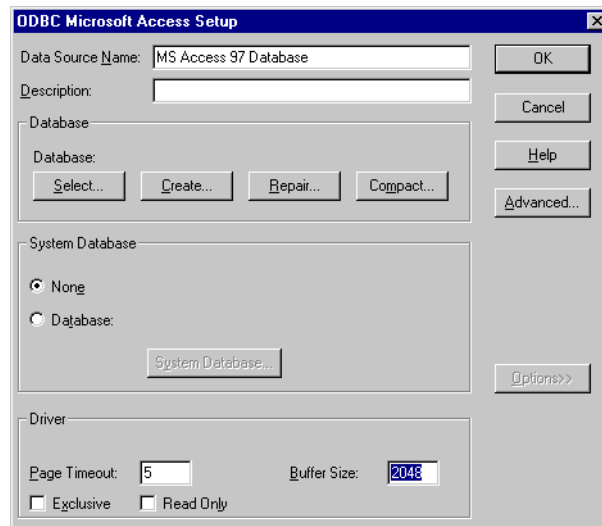


Figure 60. ODBC Setup display

9.1.2 Installing the OneWorld databases

Due to limitations in the size (maximum of 32 KB) of BLOBs in AS/400 database in OS/400 releases *prior to V4R4*, J.D. Edwards required that the OneWorld Central Objects are stored either in an Oracle or Microsoft SQL Server database.

Even though with V4R4, the DB2 UDB for AS/400 no longer has this restriction, J.D. Edwards still does not support the *native* BLOB implementation of DB2 UDB for AS/400 for OneWorld Release B733.2. However, J.D. Edwards has developed a workaround technique which breaks the BLOBs into 32 KB pieces and store them in DB2 UDB for AS/400. The workaround makes possible to use DB2 UDB for AS/400 for Central Objects. This is supported for OneWorld Release B733.2 on OS/400 V4R3 and V4R4.

Since the OneWorld Central Objects database will be stored on the AS/400 system, you no longer need to tailor creation scripts to meet the individual requirements of your installation processes.

Notes

- Central Objects on the AS/400 system require approximately 2 GB of disk space for each path code that you plan to install. You should verify that this has been included in your initial sizing.
- If you plan to install Central Objects on the AS/400, let J.D. Edwards know this when you order OneWorld. This way, an extra OneWorld Central Objects CD will be shipped with the other OneWorld CDs. This CD contains the special code required to load Central Objects on the AS/400 system and is required to complete this step.

Central Objects are loaded on the AS/400 system into new central object libraries, as shown in Table 13.

Table 13. OneWorld Central Objects data sources on the AS/400 system

Central Objects Data Source	Library name
Central Objects - CRPB733	COCRPB733
Central Objects - DEVB733	CODEVB733
Central Objects - PRODB733	COPRDB733
Central Objects - PRISTB733	COPRTB733

Loading Central Objects in DB2 UDB for AS/400

To load Central Objects in DB2 UDB for AS/400, complete the following steps:

1. Sign on the AS/400 system as security officer (QSECOFR).
2. Insert the OneWorld Central Objects CD into the AS/400 CD-ROM drive.
3. Type ADDLIBLE JDEOW on an AS/400 command line. Press Enter.
4. Type GO JDEOW/A980WMNU on an AS/400 command line. Press Enter.
5. Enter option 3 on the AS/400 command line. Press Enter.
6. Type the names of each of the Central Objects libraries (COCRPB733, CODEVB733, COPRDB733, COPRTB733) corresponding to the OneWorld environments that you are installing. Press Enter.
7. When the Central Objects are finished loading into DB2 UDB for AS/400, verify that there are 77 objects in each of these libraries using the command DSPLIB *libname*, where *libname* is the name of each of the Central Objects libraries that you loaded.

9.1.3 Planning and configuring OneWorld servers

During this phase of the OneWorld installation, the Installation Planner and Installation Workbench application are used to define and install the configuration of all required servers. Aside from the Enterprise Server, these systems may optionally include Java Application Servers, Windows Terminal Servers, or Workgroup Servers. These additional servers may be included in the plan at this

stage or added to the configuration through the creation of other plans at a later date. The choice is yours.

Environment data is entered into the Installation Planner database as defined on the pre-installation Detail Worksheet. ODBC Data Sources are created for each OneWorld data source to facilitate the required inter-system connectivity. The completed plan must be validated. In B73.3, the plan validation process is incorporated into the Installation Planner application. Once the validation process confirms that the plan is complete and that all entries are valid, it may be released. This updates the Release Master tables (in AS/400 library SYSB733), which are listed in Table 14, and the setup.inf file. The plan is then ready for use by the Installation Workbench.

Table 14. Release Master tables (library SYSB733)

Table	Description	Record count
F0085	Audit Record Y/N File - Euro	4
F0092	Library Lists - User	17
F00921	User Display Preferences	16
F00922	User Display Preferences Tag File	5
F00924	User Install Package	9
F0093	Library List Control	20
F0094	Library List Master File	6
F00941	Environment Detail - One World	6
F00942	Object Path Master File	5
F00945	Release Master	1
F00946	Release Compatibility Map	2
F00948	Release/Data Source Map	29
F00950	Security Workbench Table	0
F00960	Machine Group Identification	0
F83100	Date Title	4
F83110	Column Headings	1
F91100	Favorites Relationships and Properties	46
F91100D	Favorites - Alternative Description	0
F91300	Schedule Job Master	0
F91310	Scheduled Job Parameters	0
F91320	Job Schedule	0
F91400	Report Director Templates	11
F91410	Report Director Templates Sequence Items	34
F91420	Report Director Templates Smart Field Activation	643

Table	Description	Record count
F91430	Smart Field Template Criteria	28
F91500	Application Header for Tip of the Day	5
F91510	Tip Details for Tip of the Day	26
F96021	Software Package Build Header	0
F9603	Software Package Header	0
F96215	Software Package Build Header - History	0
F9622	Software Package Build Detail	0
F96225	Software Package Build Detail - History	0
F9631	Software Package Detail	0
F9640	CD Configuration Header File	2
F9641	CD Configuration Detail File	4
F9642	Change Table Configuration Details	0
F9643	CD Build Steps File	26
F9644	CD Director Control File	2
F9645	Software Mastering Update File	0
F9650	Machine Master	0
F9651	Machine Detail	0
F9652	Deployment Group Header	0
F9653	Deployment Group Detail Definitions	0
F9654	Deployment Location Definitions	1
F9691100	Change Table - Favorites	0
F9691400	Change Table - Report Director Template Header	0
F9691410	Change Table - Report Director Template Sequence	0
F9691420	Change Table - Smart Field Activity	0
F9691430	Change Table - Smart Field Criteria	0
F9691500	Tips Header Change Table	0
F9691510	Tip Detail Change Table	0
F9720	AutoPilot Script Repository Table	0
F972009	AutoPilot Constants Table	0
F972051	AutoPilot Script Version Table	0
F9720511	AutoPilot Script/Release Compatibility Table	0
F972052	AutoPilot Include Scripts Table	0
F972053	AutoPilot Script Object Utilization Table	0

Table	Description	Record count
F97210	AutoPilot Playback Results Header Table	0
F97211	AutoPilot Message Master Table	219
F97212	AutoPilot Next Numbers Table	0
F97214	AutoPilot Playback Results Detail Table	0
F979860	AutoPilot B732 F9860 Conversion Table	2805
F979865	AutoPilot B732 F9865 Conversion Table	7395
F98CONST	Table of Constants	48
F98DRENV	Data Replicating Environment Mapping Table	0
F98DRLOG	Data Replicating Change Log	0
F98DRPCN	Data Replicating Pending Change Notification	0
F98DRPUB	Data Replicating Publisher	33
F98DRSUB	Data Replicating Subscriber	25
F98EVDTL	Event Detail File	316
F98EVHDR	Event Header File	154
F98MOQUE	Media Object Queues	15
F98OWSEC	OneWorld Security	0
F98TMPL	Templates	0
F98VAR	Table of Variables	78
F98101	Imaging Constants	0
F9840	Installation Plan Master Table	0
F98403	Environment Plan Detail Table	0
F98405	Table Conversion Scheduler	0
F984052	Table Conversion - History Log	0
F9843	Table Conversion - JDE Scheduler	0
F986101	Object Configuration Master	1267
F98611	Data Source Master	43
F986115	Table and Data Source Sizing Table	47
F986116	MVS Table and Data Source Sizing	22
F98613	Business View Environment Server	27
F986150	Server Transfer Package Contents	0
F986151	Server Package Transfer File	0
F986152	Server Package Master	0
F98616	Printer Definition	9

Table	Description	Record count
F986161	Default Printer Table	2
F986162	Paper Definition	3
F986163	Printer Capability	59
F986164	Output Conversion	3
F986165	Printer Security	0
F986167	New Default Printer Table	0
F9882	Checkout Log Table	0
F98825	Package Deployment Scheduling	0
F98826	Package Deployment on Servers Information	0
F9883	OneWorld Network Location Table	0
F9888	Merge Log	0
F98881	Specification Merge Logging File	0
F988810	Specification Merge Tracking	0
F9889	Deployment Location Master	0
F98891	Deployment Location Path Code	0
F98892	Package Deployment Scheduling	0
F98980	Font Override by Language	12

Note: Your record counts may differ as the number of environments, locations, servers, and so on may vary.

9.1.3.1 Creating an RDBMS directory entry on the AS/400 system

You should verify the existence of a relational database directory entry on the AS/400 system. The entry name should be the same as the AS/400 system name and have a remote location name of *LOCAL. To verify whether this directory entry exists already, enter the command `WRKRDBDIRE *ALL` on an AS/400 command line, and press Enter.

If it does not exist, a relational database directory entry must be created on the AS/400 Enterprise Server using the Add RDB Directory Entry (`ADDRDBDIRE`) command. This is used to establish ODBC connections to the OneWorld data tables residing on the AS/400 server. If the directory entry is not on this list, enter the following command on an AS/400 command line, and press Enter:

```
ADDRDBDIRE RDB(sysname) RMTLOCNAME(*LOCAL) TYPE(*IP)
```

Here, *sysname* is the name of your AS/400 system.

9.1.3.2 Installation Planner

The Installation Planner is a OneWorld application that runs on the Deployment Server, automating much of the OneWorld configuration. It is comprised of a number of separate workbenches, used to collect information detailing host

system configuration, OneWorld data source, and environment configuration and deployment packages.

The Installation Planner is accessed by signing on to the OneWorld Planner environment (JDEPLAN) on the Deployment Server using the JDE user. The Installation Planner defines the following information:

- **Location Setup:** Defines the use of multiple deployment locations to OneWorld.
- **Deployment Server setup:** Defines the Deployment Server to OneWorld, its network name, and characteristics.
- **Enterprise Server setup:** Defines the Enterprise Server to OneWorld, its name, type, and characteristics.
- **Setup of other servers:** Defines the use of other servers such as a Workgroup Server, application, or Database Servers.
- **Data Source setup:** Configures the required data sources to support your configuration.
- **Environment setup:** Defines the environments and Path Codes needed to support your implementation plan.
- **Coexistent merges:** Defines which merges to run in a coexistent installation, such as the Data Dictionary.
- **Package Setup and Builds:** Defines which client workstation deployment packages are to be used. J.D. Edwards delivers pre-built standard packages that can be used during a new installation.

All of this information is stored in the Microsoft Access database JDEPLAN.MDB on the Deployment Server for use by the Installation Workbench application. After the plan is complete, you go through steps to verify and release the plan. These steps populate the appropriate tables with the configuration information. Releasing the plan creates all of the system tables on the AS/400 system in the specified system library SYSB733. At this time, only selected tables are populated.

Plan types

Two types of plans are available to the installer: a Typical Plan or a Custom Plan. A Typical Plan automatically creates OneWorld data sources based upon server information, but allows for little flexibility. A Custom Plan allows the installer to amend OneWorld data source definitions if necessary.

9.1.3.3 Creating a custom installation plan

J.D. Edwards recommends that the option to create a Custom Plan is *always* selected for AS/400 server installations where co-existence with J.D. Edwards' WorldSoftware is required, or Central Objects are stored on the AS/400 system. The flexibility of the Custom Plan is needed to make the required changes to the ODBC Data Sources.

Starting Installation Planner

To start the Installation Planner, complete the following steps:

1. Sign on to OneWorld on the Deployment Server with user `JDE` and password `JDE` in the `JDEPLAN` (planner) environment.

2. Enter GH961 in the Fast Path field. Press Enter to go to the System Installation Tools Menu.
3. Double-click **Custom Installation Plan**.
4. Click **Add** on the Work with Installation Plans display.

Entering information into Installation Planner

To enter information into the Installation Planner, complete the following steps:

1. Type the name you want to assign to your installation plan in the Name field on the Installation Planner form. Write down this name, because you will need it when running the Planner Validation report (see page 207).
2. Type a description of your plan.
3. Click **Install**.
4. Click **Coexistent with WorldSoftware** *only* if this is a OneWorld with WorldSoftware co-existence installation.
5. Type B7332 in the To Release field.
6. Click **OK**.
7. Click **OK** when prompted to Enter Location Information.

The Location Revisions form appears.

Entering plan location information

To enter plan location information, complete the following steps:

1. Type the name of your location in the Location field on the Location Revisions form.
2. Type a description for this location.
3. Since this is the primary Deployment Server, leave the Location Code and Parent Location blank.
4. Click **OK**.
5. Click **OK** when prompted to Enter Deployment Server Information.

The Deployment Server Revisions form appears.

Entering Deployment Server information

To enter Deployment Server information, complete the following steps:

1. Type the name of your Deployment Server in the machine name field on the Deployment Server Revisions form.
2. Type a description of your Deployment Server.
3. Type B7332 for the Release.
4. Type JDE for the Primary User.
5. Type 1 for the Primary Deployment Server.
6. Type \B733 for the Server Share Name.
7. Click **OK**.
8. Click **No** when prompted to add another Deployment Server.
9. Click **OK** when prompted to Enter Enterprise Server Information.

The Enterprise Server Revisions form appears.

Entering Enterprise Server information

To enter Enterprise Server information, complete the following steps:

1. Type the name of your Enterprise Server in the machine name field on the **Enterprise Server Revisions** form.
2. Type a description of your Enterprise Server.
3. Type `B7332` for the Release.
4. Type `JDE` for the Primary User.
5. Type `10` for the Host Type (AS/400).
6. Type `6008` for the Port Number if it is not correct.
7. Type the system name of your AS/400 system for the Logical Machine Name if it is not correct.
8. Type `I` for the Database Type (Client Access).
9. Type `machinename-B733 Server Map` for the Server Map Data Source if it is not correct, where *machinename* is the name of your AS/400 system.
10. Type `B7332SYS` for Installation Path if it is not correct.
11. Type the Deployment Server Name using the visual assist button.
12. Click **OK**.
13. Click **OK** when prompted to Enter Data Source Information for the Enterprise Server.

The Data Source Setup form appears for B733 Server Map.

OneWorld and ODBC Data Sources

During the creation of the Installation Plan, you are required to customize the OneWorld Data Sources and the ODBC Data Sources. The parameters required for completing the customization of these data sources are described in the following list and shown in Table 15 on page 192 and Table 16 on page 194.

Table 15 on page 192 shows the variable portions of the OneWorld Data Sources. The Library Name for each Data Source is the recommendation, based on using the J.D. Edwards standard naming convention, which is recommended. The rows are listed in alphabetical sequence. The following items list the constants for each of the OneWorld Data Sources:

- Data Source Use is always DB, except for *machine ** and *machine - Logic **, which are SVR.
- Data Source Type is always I (Client Access).
- Object Owner ID is always blank.
- Library List Name is always the name of the AS/400 Enterprise Server.
- DLL Name is always the default JDBODBC.DLL.
- Server Name is always the name of the AS/400 Enterprise Server and is always the same as the Library List Name.
- Platform is always AS400 (not RISC400).
- You always choose the default to deselect Use Table Owner.

- You always select the default Use Julian Date.
- You always select the default Decimal Shift.
- You always select the default Support for Updates.
- Database Name is always the same as the Data Source Name, except for *machine ** and *machine - Logic **, which are *machine - B733 Server Map**.

Table 15. OneWorld Data Sources

Data Source Name	Library name	Database name	OCM data source	AS/400 BLOB data source
AS/400 Common	CLTCOM **	AS/400 Common	No	No
AS/400 Common - CRP	CLTCOM **	AS/400 Common - CRP	No	No
AS/400 Common - Production	CLTCOM **	AS/400 Common - Production	No	No
AS/400 Common - Test	CLTCOM **	AS/400 Common - Test	No	No
Business Data - CRP	CRPDTA	Business Data - CRP	No	No
Business Data - CRP - DNT	CRPDTA	Business Data - CRP - DNT	No	No
Business Data - JDE	PRISTDTA	Business Data - JDE	No	No
Business Data - JDE - DNT	PRISTDTA	Business Data - JDE - DNT	No	No
Business Data - PROD	PRODDTA ***	Business Data - PROD	No	No
Business Data - PROD - DNT	PRODDTA ***	Business Data - PROD - DNT	No	No
Business Data - TEST	TESTDTA	Business Data - TEST	No	No
Business Data - TEST - DNT	TESTDTA	Business Data - TEST - DNT	No	No
Central Objects - CRPB733	COCRPB733	Central Objects - CRPB733	No	Yes
Central Objects - DEVB733	CODEVB733	Central Objects - DEVB733	No	Yes
Central Objects - PRISTB733	COPRTB733	Central Objects - PRISTB733	No	Yes
Central Objects - PRODB733	COPRDB733	Central Objects - PRODB733	No	Yes
Control Tables - CRP	CRPCTL	Control Tables - CRP	No	No
Control Tables - CRP - DNT	CRPCTL	Control Tables - CRP - DNT	No	No
Control Tables - Prod	PRODCTL	Control Tables - Prod	No	No
Control Tables - Prod - DNT	PRODCTL	Control Tables - Prod - DNT	No	No
Control Tables - Test	TESTCTL	Control Tables - Test	No	No
Control Tables - Test - DNT	TESTCTL	Control Tables - Test - DNT	No	No
Data Dictionary - B733	DDB733	Data Dictionary - B733	No	No
<i>machine *</i>	B733MAP	<i>machine - B733 Server Map *</i>	No	No
<i>machine - Logic *</i>	B733MAP	<i>machine - B733 Server Map *</i>	No	No
<i>machine - B733 Server Map *</i>	B733MAP	<i>machine - B733 Server Map *</i>	No	No
Object Librarian - B733	OLB733	Object Librarian - B733	No	No
OWJRNL	OWJRNL	OWJRNL	No	No

Data Source Name	Library name	Database name	OCM data source	AS/400 BLOB data source
System - B733	SYSB733	System - B733	Yes	No
System - B733 - DNT	SYSB733	System - B733 - DNT	No	No
Versions - CRPB733	CRPB733DNT	Versions - CRPB733	No	No
Versions - CRPB733 - DNT	CRPB733DNT	Versions - CRPB733 - DNT	No	Yes
Versions - DEVB733	DEVB733DNT	Versions - DEVB733	No	No
Versions - DEVB733 - DNT	DEVB733DNT	Versions - DEVB733 - DNT	No	Yes
Versions - PRISTB733	PRSB733DNT	Versions - PRISTB733	No	No
Versions - PRISTB733 - DNT	PRSB733DNT	Versions - PRISTB733 - DNT	No	Yes
Versions - PRODB733	PRDB733DNT	Versions - PRODB733	No	No
Versions - PRODB733 - DNT	PRDB733DNT	Versions - PRODB733 - DNT	No	Yes
<p>* <i>machine</i> is the name of your AS/400 Enterprise Server.</p> <p>** CLTCOM is the J.D. Edwards WorldSoftware default name of the common library (if you are not using the J.D. Edwards WorldSoftware defaults, substitute the name of your WorldSoftware common library).</p> <p>*** In a co-existence environment, Data Sources Business Data - PROD and Business Data - PROD - DNT map to your current WorldSoftware production library, which is CLTDTA, if you are using the J.D. Edwards recommended naming convention for WorldSoftware.</p>				

Table 16 on page 194 shows the variable portions of the ODBC Data Sources. The rows are listed in alphabetical sequence.

Note: The ODBC Data Source name must be the same as the corresponding OneWorld Data Source name.

The following items list the constants for each of the ODBC Data Sources:

- On the *Create New Data Source* form, you always select System Data Source (Applies to this machine only).
- On the *Create New Data Source* form, you always select the default Client Access ODBC Drive (32-bit).
- On the *Client Access Express ODBC Setup (32-bit)* form, on the General tab, you always select the default Client Access Express ODBC Data Source for Description.
- On the General tab, you always select the name of your AS/400 Enterprise Server for AS/400 System.
- On the Server tab, you always select the default Commit immediate (*NONE) for Commit Mode.
- On the Server tab, you always select the default 32 for Maximum field data returned.
- On the Server tab, you always select the default Search Pattern.
- On the Package tab, you always select the default Enable extended dynamic (package) support.

- On the Package tab, you always type the same value for Default Package library as you type for Default Library on the General tab.
- On the Performance tab, you always select the default Enable Lazy Close support.
- On the Performance tab, you always select the Enable Data Compression.
- On the Performance tab, you always select the default Block except if FOR UPDATE OF for Record blocking type.
- On the Performance tab, you always select the default 32 for Size.
- On the Language tab, you always select the default Sort based on HEX values for Sort type.
- On the Other tab, you always select the default Read/Write (all SQL statements allowed) for Connection type.
- On the Other tab, you always select the default OS/400 object description for Object description type.
- On the Other tab, you always select the default Scrollable unless rowset is 1 for Scrollable cursor.
- On the Format tab, you always select the System Naming Convention (*SYS).

Table 16. ODBC Data Sources

Data Source Name	Default library name	Default package library	Do Not Translate CCSID 65535	Translate CCSID 65535
AS/400 Common	CLTCOM **	CLTCOM **	No	Yes
AS/400 Common - CRP	CLTCOM **	CLTCOM **	No	Yes
AS/400 Common - Production	CLTCOM **	CLTCOM **	No	Yes
AS/400 Common - Test	CLTCOM **	CLTCOM **	No	Yes
Business Data - CRP	CRPDTA	CRPDTA	No	Yes
Business Data - CRP - DNT	CRPDTA	CRPDTA	Yes	No
Business Data - JDE	PRISTDTA	PRISTDTA	No	Yes
Business Data - JDE - DNT	PRISTDTA	PRISTDTA	Yes	No
Business Data - PROD	PRODDTA ***	PRODDTA ***	No	Yes
Business Data - PROD - DNT	PRODDTA ***	PRODDTA ***	Yes	No
Business Data - TEST	TESTDTA	TESTDTA	No	Yes
Business Data - TEST - DNT	TESTDTA	TESTDTA	No	No
Central Objects - CRPB733	COCRPB733	COCRPB733	Yes	No
Central Objects - DEVB733	CODEVB733	CODEVB733	Yes	No
Central Objects - PRISTB733	COPRTB733	COPRTB733	Yes	No
Central Objects - PRODB733	COPRDB733	COPRDB733	Yes	No
Control Tables - CRP	CRPCTL	CRPCTL	No	Yes
Control Tables - CRP - DNT	CRPCTL	CRPCTL	Yes	No
Control Tables - Prod	PRODCTL	PRODCTL	No	Yes

Data Source Name	Default library name	Default package library	Do Not Translate CCSID 65535	Translate CCSID 65535
Control Tables - Prod - DNT	PRODCTL	PRODCTL	Yes	No
Control Tables - Test	TESTCTL	TESTCTL	No	Yes
Control Tables - Test - DNT	TESTCTL	TESTCTL	Yes	No
Data Dictionary - B733	DDB733	DDB733	No	Yes
<i>machine</i> *	B733MAP	B733MAP	Yes	No
<i>machine</i> - Logic *	B733MAP	B733MAP	Yes	No
<i>machine</i> - B733 Server Map *	B733MAP	B733MAP	Yes	No
Object Librarian - B733	OLB733	OLB733	No	Yes
OWJRNL	OWJRNL	OWJRNL	Yes	No
System - B733	SYSB733	SYSB733	No	Yes
System - B733 - DNT	SYSB733	SYSB733	Yes	No
Versions - CRPB733	CRPB733DNT	CRPB733DNT	No	Yes
Versions - CRPB733 - DNT	CRPB733DNT	CRPB733DNT	Yes	No
Versions - DEVB733	DEVB733DNT	DEVB733DNT	No	Yes
Versions - DEVB733 - DNT	DEVB733DNT	DEVB733DNT	Yes	No
Versions - PRISTB733	PRSB733DNT	PRSB733DNT	No	Yes
Versions - PRISTB733 - DNT	PRSB733DNT	PRSB733DNT	Yes	No
Versions - PRODB733	PRDB733DNT	PRDB733DNT	No	Yes
Versions - PRODB733 - DNT	PRDB733DNT	PRDB733DNT	Yes	No
<p>* <i>machine</i> is the name of your AS/400 Enterprise Server</p> <p>** CLTCOM is the J.D. Edwards WorldSoftware default name of the common library (if you are not using the J.D. Edwards WorldSoftware defaults, substitute the name of your WorldSoftware common library).</p> <p>*** In a co-existence environment, Data Sources Business Data - PROD and Business Data - PROD - DNT map to your current WorldSoftware production library, which is CLTDTA, if you are using the J.D. Edwards recommended naming convention for WorldSoftware.</p>				

Entering Enterprise Server data source information

The Installation Workbench prompts the user to create each of the ODBC Data Sources in the order in which these data sources are listed in the plan. Complete the following steps:

1. Verify the B733 Server Map source information on the Data Source Setup form. It should look similar to the display in Figure 61 on page 196, except that your AS/400 system name appears instead of DENAS2. Make any necessary changes, and click **Advanced** button.

Figure 61. OneWorld Data Source setup

2. Verify the B733 Server Map advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 62, except that your AS/400 system name appears instead of DENAS2. Make any necessary changes, and click **OK**. The Create New Data Source form appears. Do *not* click Cancel as described in the OneWorld Installation Guide (AS/400 Systems) as the ODBC Data Source information requires changes.

Figure 62. OneWorld Data Source revisions

3. Click **System Data Source** on the Create New Data Source form as shown in Figure 63. Click **Next**. The Create New Data Source form to select the correct data source driver appears, as shown in Figure 64. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, as shown in Figure 65.

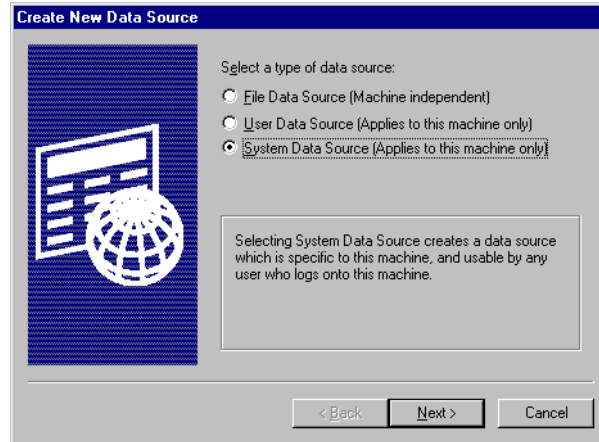


Figure 63. Create New OneWorld Data Source (Part 1 of 3)

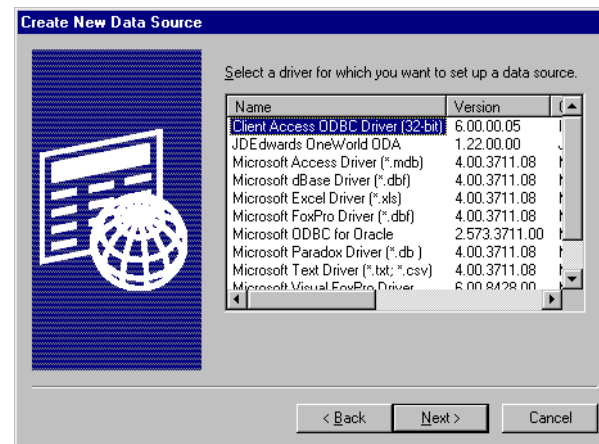


Figure 64. Create New OneWorld Data Source (Part 2 of 3)

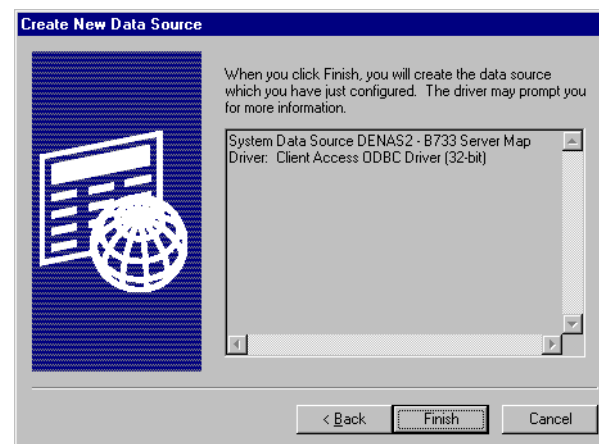


Figure 65. Create New OneWorld Data Source (Part 3 of 3)

4. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear similar to the example in Figure 66 on page 198, except your AS/400 system name is substituted for DENAS2.

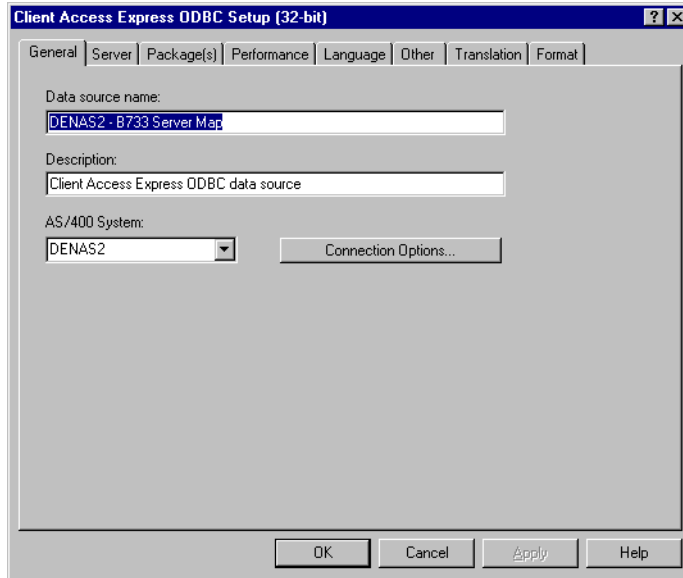


Figure 66. Client Access Express ODBC Setup (32-bit) (Part 1 of 6)

5. Click the **Server** tab.
6. Type B733MAP for Default Libraries as shown in Figure 67.

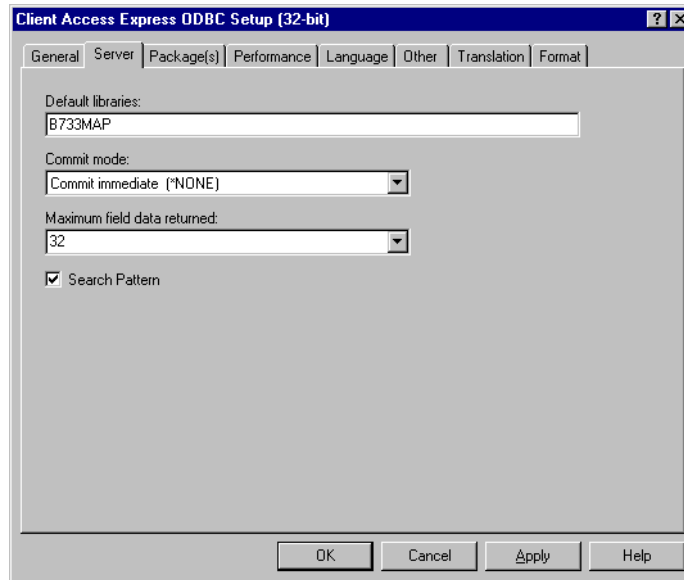


Figure 67. Client Access Express ODBC Setup (32-bit) (Part 2 of 6)

7. Click the **Package(s)** tab.
8. Type B733MAP for Default package library as shown in Figure 68.

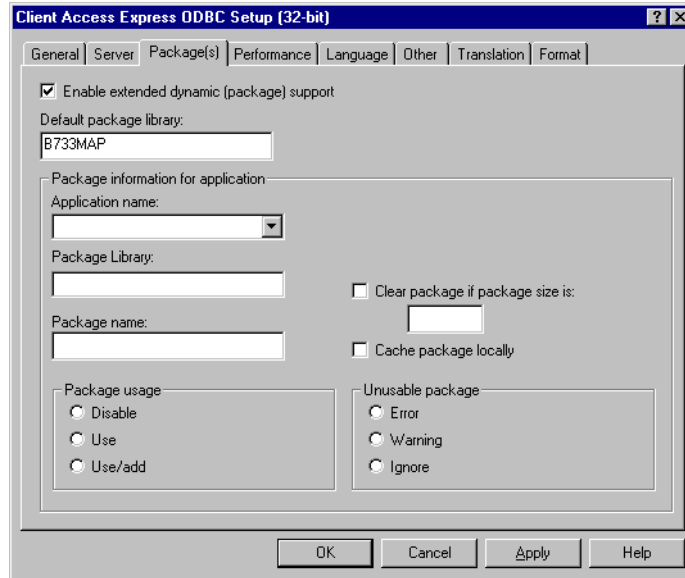


Figure 68. Client Access Express ODBC Setup (32-bit) (Part 3 of 6)

9. Click the **Performance** tab.

10. Select **Enable Lazy Support**. Select **Enable Data Compression** as shown in Figure 69.

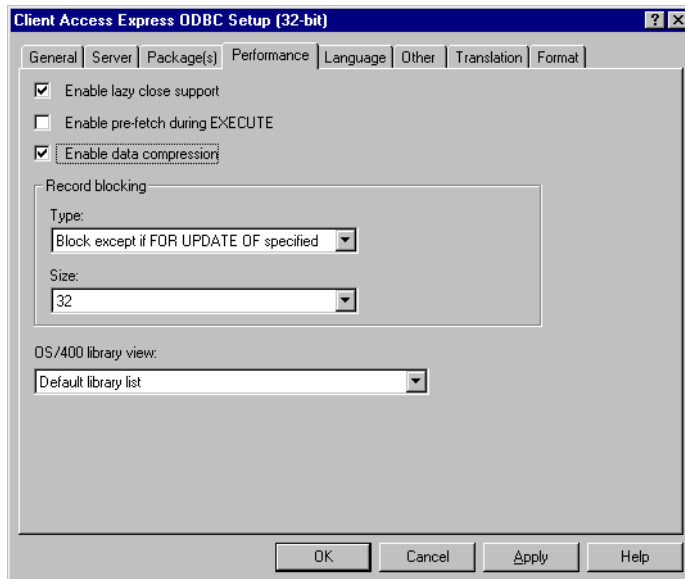


Figure 69. Client Access Express ODBC Setup (32-bit) (Part 4 of 6)

11. Click the **Translation** tab.

12. Select **Do not translate CCSID 65535** as shown in Figure 70 on page 200.

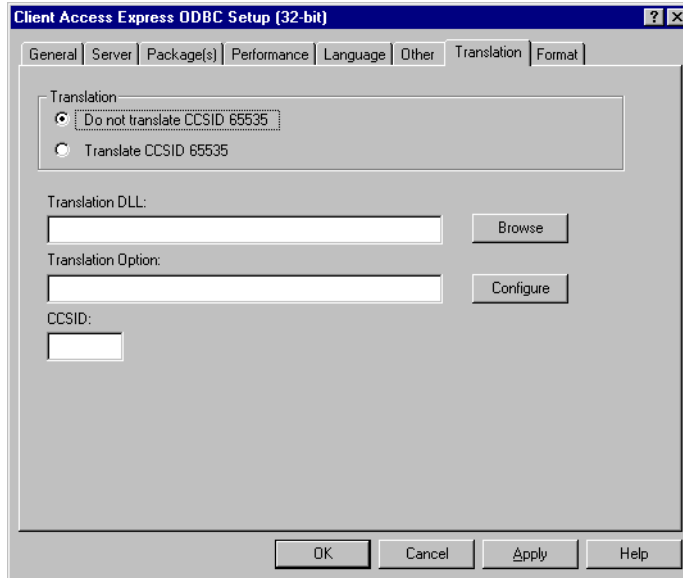


Figure 70. Client Access Express ODBC Setup (32-bit) (Part 5 of 6)

13. Click the **Format** tab.

14. Select **System naming convention (*SYS)** for Naming convention as shown in Figure 71.

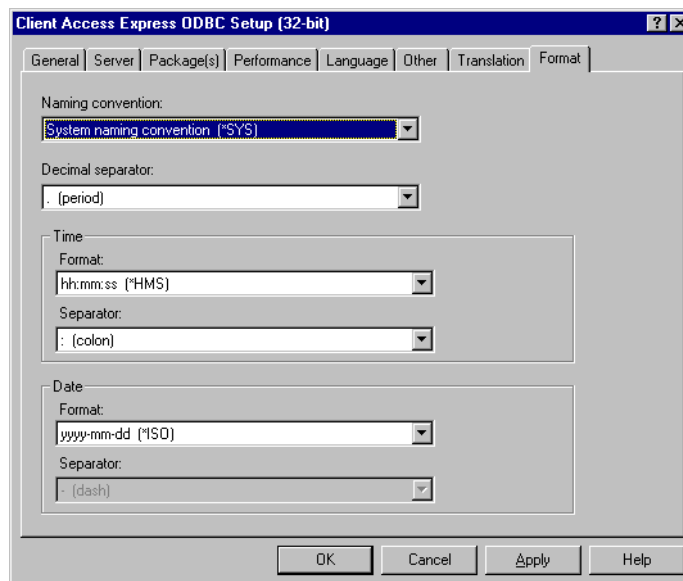


Figure 71. Client Access Express ODBC Setup (32-bit) (Part 6 of 6)

15. Click **Apply**.

16. Click **OK**.

17. If you receive an ODBC Failure message, click **Cancel**, since the ODBC Data Source may already exist.

18. If you receive the "Data source already exists Overwrite?" prompt, click **Yes**, to replace the original ODBC settings.

19. The Data Source Setup form for the B733 Server Map re-appears, similar to the example in Figure 61 on page 196. Click **OK** to continue.
20. Click **No** when prompted to add another Enterprise Server. The Machines/Servers Types form appears.

Defining machine/server types

To specify machine/server types, complete the following steps.

1. Deselect **Data Server**.

Note

This prompt states that a data server must be defined for your Central Objects when your Enterprise Server is an AS/400 system. However, this is only the case when the Central Objects are *not* installed in DB2 UDB for AS/400, as you are doing.

2. Deselect **JAS Server**, because you are not configuring a Java Application Server.
3. Deselect **WTS Server**, because you are not configuring a Windows Terminal Server.

Your form should appear similar to the example in Figure 72.

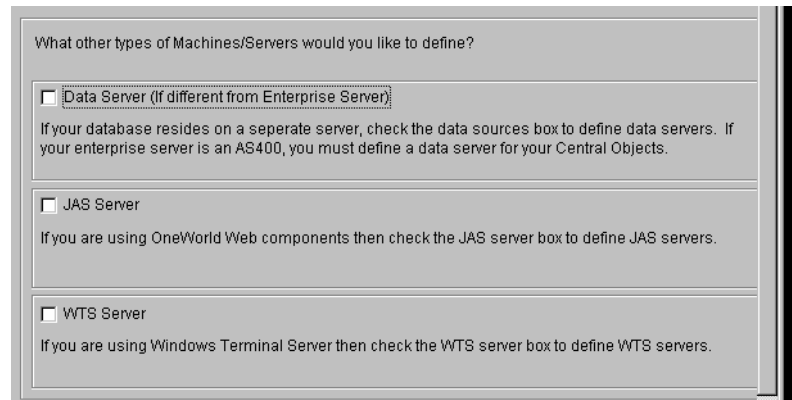


Figure 72. OneWorld machine/server types

4. Select **OK** to continue.

Entering shared data source setup information

To specify the shared data source setup information, complete the following steps:

1. Select **OK** when prompted to enter Data Source Information for the shared data sources. The Data Source Setup form for Data Source System - B733 appears.
2. Verify the System - B733 source information on the Data Source Setup form. It should look similar to the example in Figure 61 on page 196 except the Data Source Name is now System - B733 and Data Library Name is now SYSB733. Make any necessary changes, and click the **Advanced** button.
3. Verify the System - B733 advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 62 on page 196 except the

Library Name is now SYSB733 and the Database Name is now System - B733. Select **OCM Data Source**. Make any other necessary changes, based on Table 15 on page 192. Click **OK**. The Create New Data Source form appears.

4. Click **System Data Source** on the Create New Data Source form and you should see a form similar to Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the example in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.
5. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except the Data source name is now System - B733.
6. Click the **Server** tab.
7. Type `SYSB733` for Default Libraries. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
8. Click the **Package(s)** tab.
9. Type `SYSB733` for Default package library. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
10. Click the **Performance** tab.
11. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as shown in Figure 69 on page 199.
12. Click the **Translation** tab.
13. Select **Translate CCSID 65535**. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
14. Click the **Format** tab.
15. Select **System naming convention (*SYS)** for Naming convention. The information should appear the same as shown in Figure 69 on page 199.
16. Click **Apply**.
17. Click **OK**.
18. Click **Yes** in reply to "Data source already exists Overwrite?".
19. The Data Source Setup form for System - B733 re-appears, similar to the example in Figure 61 on page 196. Click **OK** to continue. The Data Source Setup form for Data Source Object Librarian - B733 appears.
20. Verify the Object Librarian - B733 source information on the Data Source Setup form. It should look similar to the example in Figure 61 on page 196, except the Data Source Name is now Object Librarian - B733 and Data Library Name is now OLB733. Make any necessary changes, and click **Advanced** button.
21. Verify the Object Librarian - B733 advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 62 on page 196, except the Library Name is now OLB733 and the Database Name is now Object Librarian - B733. Make any necessary changes, and click **OK**. The Create New Data Source form appears.

22. Click **System Data Source** on the Create New Data Source form. You should see a form similar to the example in Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the example in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.
23. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except the Data source name is now Object Librarian - B733.
24. Click the **Server** tab.
25. Type `OLB733` for Default Libraries. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
26. Click the **Package(s)** tab.
27. Type `OLB733` for Default package library. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
28. Click the **Performance** tab.
29. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as the example in Figure 69 on page 199.
30. Click the **Translation** tab.
31. Select **Translate CCSID 65535**. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
32. Click the **Format** tab.
33. Select **System naming convention (*SYS)** for Naming convention. Otherwise, the information should appear the same as shown in the display in Figure 69 on page 199.
34. Click **Apply**.
35. Click **OK**.
36. If the "Data source already exists Overwrite?" prompt appears, click **Yes**.
37. The Data Source Setup form for Object Librarian - B733 re-appears, similar to Figure 61 on page 196. Click **OK** to continue. The Data Source Setup form for Data Source Data Dictionary - B733 appears.
38. Verify the Data Dictionary - B733 source information on the *Data Source Setup* form. It should look similar to the example in Figure 61 on page 196, except the Data Source Name is now Data Dictionary - B733 and Data Library Name is now DDB733. Make any necessary changes, based on Table 15 on page 192. Click the **Advanced** button.
39. Verify the Data Dictionary - B733 advanced settings on the Data Source Revisions form. It should look similar to the display in Figure 62 on page 196, except the Library Name is now DDB733 and the Database Name is now Data Dictionary - B733. Make any necessary changes, and click **OK**. The Create New Data Source form appears.
40. Click **System Data Source** on the Create New Data Source form and you should see a form similar to the display in Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the example in Figure 64 on page 197. Select **Client**

Access ODBC (32-bit). Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the display in Figure 65 on page 197.

41. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except the Data source name is now Data Dictionary - B733.
42. Click the **Server** tab.
43. Type `DDB733` for Default Libraries. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
44. Click the **Package(s)** tab.
45. Type `DDB733` for Default package library. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
46. Click the **Performance** tab.
47. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as shown in Figure 69 on page 199.
48. Click the **Translation** tab.
49. Select **Translate CCSID 65535**. Otherwise, the information should appear like the example in Figure 70 on page 200.
50. Click the **Format** tab.
51. Select **System naming convention (*SYS)** for Naming convention. Otherwise, the information should appear the same as shown in Figure 69 on page 199.
52. Click **Apply**.
53. Click **OK**.
54. If the “Data source already exists Overwrite?” prompt appears, click **Yes**.
55. The Data Source Setup form for Data Dictionary - B733 re-appears, similar to what you see in Figure 61 on page 196. Click **OK** to continue.

Setting up environments and environment data sources

To specify the environment parameters, complete the following steps:

1. After the Environment Selection form appears, deselect **Default Environments**.
2. Deselect **Default Data Load**.
3. Deselect **Default Advanced Parameters**.
4. Select **Default Languages**, unless you are installing multiple languages.
The form should look similar to the example in Figure 73.

Default Environments
To configure all valid environments for your installation, check the Default Environments box. Otherwise uncheck the Default Environments box to select individual environments.

Default Data Load
To utilize JD Edwards data load options default for loading environment data, check the Default Data Load box. Otherwise uncheck the Default Data Load box to override the data load options.

Default Advanced Parameters
To utilize JD Edwards advanced parameters defaults for upgrade, update and co-existent merges, check the Default Advanced Parameters box. Otherwise uncheck the Default Advanced Parameters box to override the advanced parameters options.

Default Languages
To install all valid languages for each selected environment, check the Default Languages box. Otherwise uncheck the Default Languages box to select languages.

Figure 73. OneWorld Environment selections

5. Click **OK**. The Select Environments form appears.
6. Repeat the following steps for each environment that you are defining:
 - a. Double-click the row corresponding to the environment you want to define.
The Data Load Parameters form appears.
 - b. Select **Load Production Data** for all environments except Pristine environment.
 - c. Select **Load Demo Data** for Pristine environment only.
 - d. The form should look similar to the example in Figure 74. Click **OK**. The Advanced Parameters form appears.

Do Not Load Data Load Production Data Load DemoData

Data Load Setup

Environment	
PRD733	B733 Production Environment

Figure 74. OneWorld Advanced Parameters

- e. *Only* if you are installing OneWorld in a co-existence environment with WorldSoftware, select **Coexistent Environment**, and select **Coexistent Merges**. Deselect **Coexistent Environment**, and deselect **Coexistent Merges** *always* for the Pristine environment.
- f. If necessary, deselect **Upgrade Environment** because this is a new installation.
- g. Click **OK**.
- h. Click **OK** when the enter Data Source Information prompt appears. J.D. Edwards recommends that you do *not* select defaults for the AS/400 system. Depending on the environment you select and whether you select

co-existence, you now set up the environment data sources for any or all of the data sources: OWJRNL, Control Tables, AS/400 Common, Central Objects, Business Data, and Versions. Complete the following steps to set up the environment data sources for each environment:

- i. After the Data Source Setup form for the next Data Source appears, verify the source information on the Data Source Setup form. It should look similar to the example in Figure 61 on page 196, except for the Data Source Name and Data Library Name. Make any necessary changes, based on Table 15 on page 192. Click the **Advanced** button.
- ii. Verify the advanced settings on the Data Source Revisions form. It should look similar to what is shown in Figure 62 on page 196 except for the Library Name and the Database Name. Make any necessary changes and click **OK**. The Create New Data Source form appears.
- iii. Click **System Data Source** on the Create New Data Source form. You should see a form similar to the one in Figure 63 on page 197.
- iv. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the one shown in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.
- v. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except for the Data source name.
- vi. If you previously defined this ODBC Data Source, you may click **Cancel**. Return to step 1 with the next OneWorld data source.
- vii. Click the **Server** tab.
- viii. Enter the corresponding library name for this data source for Default Libraries, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
- ix. Click the **Package(s)** tab.
- x. Enter the corresponding library name for this data source for the Default package library, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
- xi. Click the **Performance** tab.
- xii. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as shown in Figure 69 on page 199.
- xiii. Click the **Translation** tab.
- xiv. Select either **Do Not Translate CCSID 65535** or **Translate CCSID 65535**, corresponding to what is required for this data source, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
- xv. Click the **Format** tab.
- xvi. Select **System naming convention (*SYS)** for Naming convention. The information should appear the same as shown in Figure 69 on page 199.

- xvii. Click **Apply**.
 - xviii. Click **Yes** in reply to “Data source already exists Overwrite?”.
 - xix. Click **OK**.
 - xx. If you receive an ODBC Failure message, click **Cancel**, since the ODBC Data Source may already exist.
 - xxi. The Data Source Setup form for this data source re-appears, similar to the example in Figure 61 on page 196. Click **OK** to continue.
 - xxii. If the Data Source Setup form for the next data source appears, repeat this series of steps for configuring environment data sources for the next data source.
 - i. The Select Environments form re-appears, showing *Selected* for the environment you set up. If you have another environment that you want to set up, repeat this series of steps for the next environment you want to define.
7. When you are finished defining all of your OneWorld environments, click **Close** on the Select Environments form.
 8. Click **No** when prompted to add another location.
 9. Click **OK** when you see the message indicating that the plan has been finalized. The Work With Batch Versions - Available Versions form appears for Batch Application R9840B (Planner Validation Report).

Plan Validation

The Plan Validation process is included as an integral part of the Installation Planner application in Release B73.3, and is initiated automatically. This process ensures the plan contains all of the necessary information. The Installation Planner Validation Report (R9840B) should display “Record Validated” in the far right-hand column for each record. If this is not the case, revise the plan until all records are validated prior to finalization.

Running the Planner Validation report

To run the Planner Validation report, complete the following steps:

1. Double-click Version **XJDE0001** to initiate the batch application.
2. Select **Data Selection** on the Version Prompting form.
3. Click **Submit**.
4. Click “**TESTPLAN**” on the Data Selection form.
5. Double-click <**Literal**>.
6. Enter your custom installation plan name assigned when you first created your custom plan in 9.1.3.3, “Creating a custom installation plan” on page 189. Your display should look similar to the example in Figure 75 on page 208, except for your plan name.

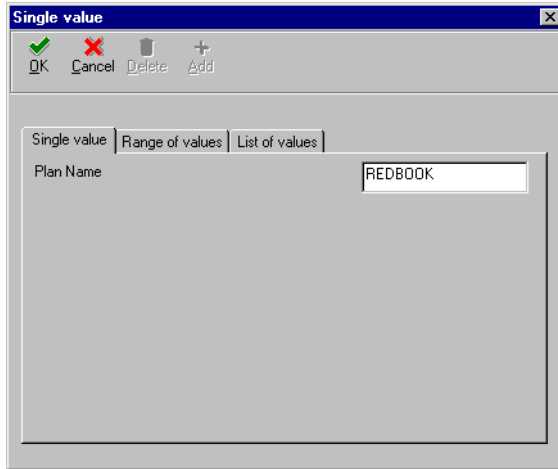


Figure 75. OneWorld Planner Validation Report data selection (Part 1 of 3)

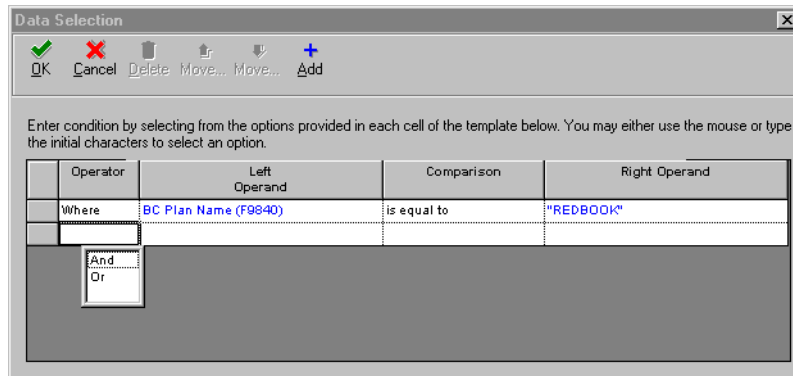


Figure 76. OneWorld Planner Validation Report data selection (Part 2 of 3)

7. Click **OK**.
8. The Data Selection form re-appears with your plan name showing in the Right Operand, similar to the example in Figure 76. Click **OK**.

The Processing Options Extra Info tab appears. Chapter 7 of the *OneWorld Installation Reference Guide (All Systems)* describes, in detail, the available processing options and suggests which settings may be suitable.

9. Type **Y** for Additional data source information flag.
10. Type **D** for OCM Information Flag.
11. Type **N** for OCM Differences Flag.
12. Type **Y** for Path Code Information Flag.
13. Type **30** for Update the Plan Status.

Figure 77. OneWorld Planner Validation Report data selection (Part 3 of 3)

Your display should look like the example in Figure 77. Click **OK**.

14. The Report Output Destination form appears. Select **On Screen**.

15. Click **OK**.

16. Review the resulting R9840B Validation Planner Report to verify that your installation plan is correct. Validation Results of Record Validated and Duplicate Entry are normal. Other Validation results should be investigated before continuing with the Installation Workbench.

Other Validation results

In the Machines section of the R9840B Validation Report, a Validation Result `FTP Server has not been setup` may appear for Machine Description `AS/400 Enterprise Server`. If you can successfully use FTP in both directions between the Deployment Server and `AS/400 Enterprise Server`, this error can be ignored. It does not cause any further issues that the Detail Status remains at 20. This information can be found in the Breaking News item from the J.D. Edwards Knowledge Garden (<http://www.jdedwards.com>).

In the Specifications Table Merge section of the R9840B Validation Report, a Validation Result `Env. Plan Match` may appear for Environment `TST733`. This is an incorrect message, since `TST733` and `CRP733` share a common data path. If this same object (for example, `F98750`) and same target Data Source (for example `Central Objects - CRPB733`) appear with a Validation Result `Record Validated` for `CRP733`, this message can be ignored. It does not cause any further issues that the Detail Status remains at 20.

17. On the Work With Installation Plans form, expand your plan and each part of the plan to review.

Running Release Master

After you create the installation plan, you must release it to initialize the System data source and create the System tables. Before running the release master, you must create a user profile JDE on the AS/400 system with at least special authorities `*JOBCTL` and `*ALLOBJ`. These special authorities are required so the

Installation Workbench can run successfully. To create this JDE user profile on the AS/400 system, complete the following steps:

1. Sign on to the Deployment Server as Administrator.
2. Click **Start->Programs->IBM AS/400 Client Access Express->AS/400 Operations Navigator**. This starts AS/400 Operations Navigator for accessing the AS/400 from the Deployment Server.
3. If necessary, expand **My AS/400 Connections** to see your AS/400 Enterprise Server.
4. Expand your AS/400 Enterprise Server.
5. When prompted for Signon to AS/400, type QSECOFR for AS/400 User ID and the corresponding password for Password. Click **OK**.
6. Expand **Users and Groups**.
7. Right-click **All Users**.
8. Click **New User**.
9. Type JDE for User name and an appropriate description.
10. For Password, type the same password used for user JDE on the Deployment Server.
11. Deselect **User must change password at next logon**.
12. Click **Capabilities**.
13. If necessary, select the **Privileges** tab.
14. Select **All object access**.
15. Select **Job control**.
16. Click **OK**.
17. Click **Add** on the New User display.

To run the Release Master, complete the following steps:

1. If you are not still signed onto OneWorld with the Work with Installation Plans form appearing, complete the following steps, as necessary:
 - a. Sign on to OneWorld on the Deployment Server with user JDE in the JDEPLAN environment.
 - b. Type GH961 in Fastpath and click **Enter**.
 - c. Double-click **Custom Installation Plan**.
2. Double-click your custom installation plan.
3. If the Exit Bar is not displayed, click **Preferences->Exit Bar**.
4. From the Form menu on the Exit Bar, click **Release**.
The Create New Data Source form appears.
5. Click **System Data Source** on the Create New Data Source form. You should see a form similar to the display in Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the example in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.

6. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except the Data source name is now System-B733.
7. Click **Cancel**, since you previously defined the ODBC Data Source for the System - B733 data source.
The Create New Data Source form appears.
8. Click **System Data Source** on the Create New Data Source form. You should see a form similar to the display in Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, as shown in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears. This is similar to the display in Figure 65 on page 197, except the System Data Source is now System - B733 - DNT.
9. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except the Data source name is now System - B733 - DNT.
10. Click the **Server** tab.
11. Type `SYSB733` for Default Libraries. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
12. Click the **Package(s)** tab.
13. Type `SYSB733` for Default package library. Otherwise, the information should appear as shown in Figure 68 on page 199.
14. Click the **Performance** tab.
15. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear as shown in Figure 69 on page 199.
16. Click the **Translation** tab.
17. Select **Do not translate CCSID 65535**. The information should appear the same as shown in Figure 70 on page 200.
18. Click the **Format** tab.
19. Select **System naming convention (*SYS)** for Naming convention. The information should appear the same as shown in Figure 69 on page 199.
20. Click **Apply**.
21. Click **Yes** in reply to "Data source already exists Overwrite?".
22. Click **OK** in reply to the message that the release master process has completed.
23. Click **Close**.

Defining OneWorld AS/400 Common Data Sources (co-existence only)

If you are installing a OneWorld and WorldSoftware coexistence environment, you define at this time your AS/400 and OneWorld Client Access ODBC Common Data Sources.

1. Double-click **Advanced Operations** on the System Installation Tools menu (GH961).
2. Double-click **Database Data Sources**. You should see a display similar to the example in Figure 78 on page 212.



Figure 78. OneWorld machine selection

3. Click **LOCAL OneWorld Planner - B733**. Click **Select**.

The Work With Data Sources form appears.

4. If necessary, type **DB** for Data Source Use. Click **Find**.

The Work With Data Sources form now looks similar to the example in Figure 79.

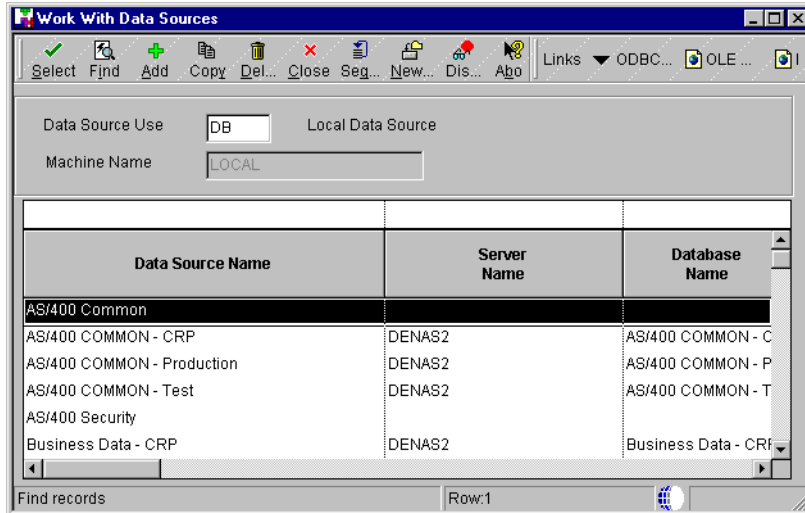


Figure 79. OneWorld Work With Data Sources

5. Complete each of the following steps for each of the listed AS/400 Common Data Sources, if this is a OneWorld and WorldSoftware co-existence installation:

- a. Click the next undefined AS/400 Common Data Source. Click **Select**.
- b. Type the appropriate information on the Data Source Revisions form, corresponding to the selected data source name in Table 15 on page 192. It should look similar to the example in Figure 62 on page 196, except the Library Name is now **CLTCOM** (or your assigned Common library name),

and the Database Name is now the same as your selected Data Source Name. Click **OK**. The Create New Data Source form appears.

- c. Click **System Data Source** on the Create New Data Source form. You should see a form similar to the one shown in Figure 63 on page 197. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**. The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.
- d. Click **Finish**. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as the example in Figure 66 on page 198, except the Data source name is now the same as your selected Data Source Name.
- e. Click the **Server** tab.
- f. Enter `CLTCOM` (or your assigned Common library name) for Default Libraries, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
- g. Click the **Package(s)** tab.
- h. Enter `CLTCOM` (or your assigned Common library name) for Default package library, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
- i. Click the **Performance** tab.
- j. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as shown in Figure 69 on page 199.
- k. Click the **Translation** tab.
- l. Select **Translate CCSID 65535**, corresponding to what is required for this data source, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
- m. Click the **Format** tab.
- n. Select **System naming convention (*SYS)** for Naming convention. Otherwise, the information should appear the same as shown in Figure 69 on page 199.
- o. Click **Apply**.
- p. If the “Data source already exists Overwrite?” prompt appears, click **Yes**.
- q. Click **OK**.

The Work With Data Sources display now re-appears, similar to the display in Figure 79 on page 212. If there are more AS/400 Common Data Sources to define, repeat this series of steps.

6. Click **Close** after all AS/400 Common Data Sources have been defined. Log off OneWorld.

9.1.3.4 Installation Workbench

The Installation Workbench application executes the plan created in the Installation Planner phase, by copying tables from the Planner environment into the data sources of the defined environments on the appropriate servers.

Installation Workbench consists of several separate workbench functions, which include:

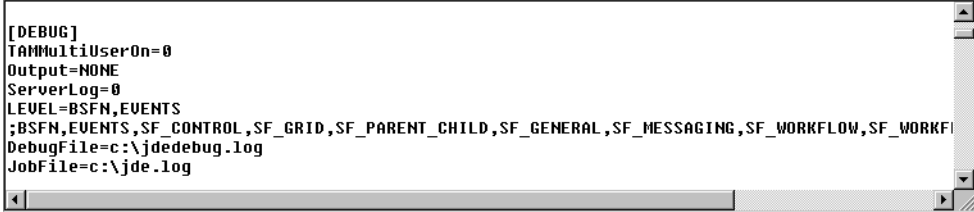
- **Location Workbench:** Updates the System - B733 data source based on the deployment locations defined in the plan.
- **Data Source Workbench:** Copies all data sources defined in the planner from the F98611 Data Source Master and F986115 Table and Data Source Sizing tables in the planner data source into the System - B733 data source. This workbench function also walks you through configuring the necessary ODBC Data Sources.
- **Environment Workbench:** Copies the F0093, F0094, F00941, and F00942 environment information tables and the F986101 Object Configuration Manager table into the System - B733 data source. Depending on the plan's settings, it also runs batch applications, which create tables and indexes for the required environments.

Also, depending upon your plan's setting, coexistent merges may be launched at this time.
- **Host Workbench:** Copies the F00053, F000531, and F000532 host configuration tables into the System - B733 data source. Then, it populates the F986101, F98611, and F986115 Server Map data source tables with all valid environment information.
- **Package Workbench:** Transfers the F9885, F9886, and F9887 package information tables to the System - B733 data source.

Before commencing the Installation Workbench procedure, perform the tasks described in the following sections.

Switching off debugging

We recommend that you switch off debugging during the execution of the Installation Workbench. If it is switched on, it degrades the performance of the process creates a large log file that wastes disk space. To ensure that this is off, check the jde.ini file located in the \winnt directory. The OUTPUT statement in the DEBUG section should equal NONE (see Figure 80).



```
[DEBUG]
TAMMultiUserOn=0
Output=NONE
ServerLog=0
LEVEL=BSFN,EVENTS
;BSFN,EVENTS,SF_CONTROL,SF_GRID,SF_PARENT_CHILD,SF_GENERAL,SF_MESSAGING,SF_WORKFLOW,SF_WORKFI
DebugFile=c:\jdedebug.log
JobFile=c:\jde.log
```

Figure 80. jde.ini File (DEBUG section)

Verifying JDE passwords

You should verify that user profile JDE is on your AS/400 system. If it is not, you should create it now so that the Installation Workbench can run successfully. You should also verify that it has the same password as you gave to the user profile JDE on the Deployment Server. Refer to "Running Release Master" on page 209.

The JDE user profile password is cached on the Deployment Server. This password is used to access the JDE user profile on the AS/400 server during the Installation Workbench process. Therefore, the passwords must be synchronized

on both systems to allow successful ODBC connections to be established. If the passwords do not match on the two systems, each attempt to initialize an ODBC connection causes a database signon window to be issued that asks for the correct AS/400 password to be entered.

Verify that the AS/400 user profile JDE has special authority *JOBCTL and *ALLOBJ. These special authorities are required for the Installation Workbench to run successfully.

Verify that the object authority *CHANGE is assigned to *PUBLIC for the AS/400 user profile JDE. This is required later for OneWorld client workstation users to successfully submit UBEs to run on the AS/400 server. If necessary, the object authority can be changed using the Edit Object Authority (EDTOBJAUT) command.

Verifying CCSID

The *OneWorld Installation Guide (AS/400 Systems)* incorrectly states that the user's *preferred* language character set can be specified for the AS/400 user profile JDE during execution of the Installation Workbench.

Attention

Currently, *only* CCSIDs 37 and 65535 are suitable for use during the installation phase. Therefore, the JDE AS/400 user profile must be checked and amended if necessary.

For more information on the effects of CCSID and using languages other than English within OneWorld, refer to Chapter 17, "National language support" on page 471.

Starting Installation Workbench and configuring your locations

To start the Installation Workbench, complete the following steps:

1. Log onto J.D. Edwards OneWorld on the Deployment Server as user JDE and planner environment JDEPLAN.
2. If necessary, type GH961 on Fastpath and press Enter to display the System Installation Tools menu.
3. Double-click **Installation Workbench**.
4. Double-click your installation plan.
5. The Location Workbench form appears. Verify that all of the locations for your plan are listed before continuing.
6. Click **Configure** on the Form menu.

OneWorld updates the Detail Status (60) and Status Description (Installed). Your display should look similar to the example in Figure 81 on page 216.

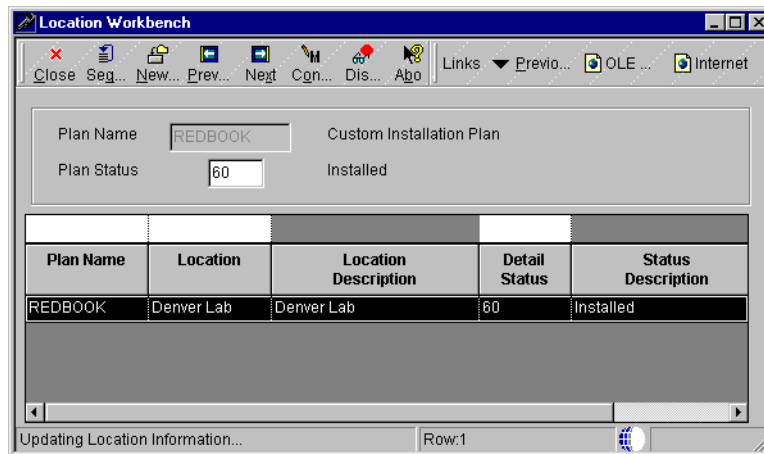


Figure 81. OneWorld Location Workbench

7. Click **Next**. The Data Source Workbench form appears listing the required OneWorld data sources for your installation plan.

Configuring your data sources

You need to define all ODBC Data Sources corresponding to all OneWorld data sources except the OneWorld Local data sources.

Each ODBC Data Source, which you did not define previously while you were creating the installation plan, can be selected by selecting Configure on the Row menu on the Data Source Workbench. However, there is a risk that you may miss a required data source.

The following procedure outlines the steps using the Form menu on the Data Source Workbench to configure the ODBC Data Sources for every OneWorld data source (except the OneWorld Local data sources). When OneWorld prompts for the setup of an ODBC Data Source that you already defined, you can cancel or redefine this data source.

1. Verify that all of the required OneWorld data sources for your plan are listed on the Data Source Workbench form.
2. Click **Configure** on the Form menu (not the Row menu).
3. Repeat the following steps to configure the ODBC Data Sources corresponding to each of the listed OneWorld data sources:
 - a. After the Create New Data Source form appears, click **System Data Source** on the Create New Data Source form. You should see a form similar to the example in Figure 63 on page 197.
 - b. Click **Next**. The Create New Data Source form to select the correct data source driver appears, similar to the example in Figure 64 on page 197. Select **Client Access ODBC (32-bit)**. Click **Next**.
 - c. If you see the Create New Data Source form for the data source to which you want to save this connection, similar to the display in Figure 82, click **Next**.

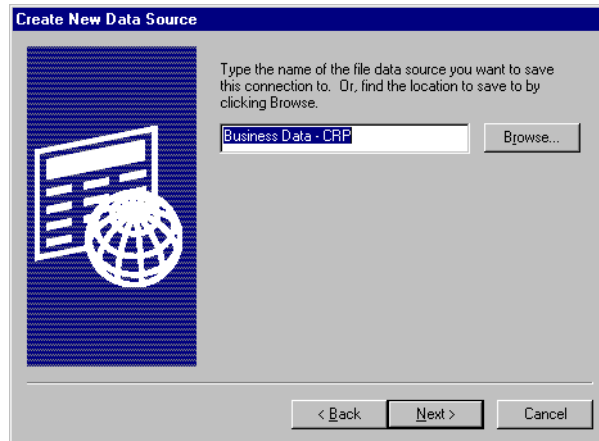


Figure 82. Create New OneWorld Data Source

The Create New Data Source form to verify the creation of the configured data source appears, similar to the example in Figure 65 on page 197.

- d. Click **Finish**.
- e. If you see the message shown in Figure 83, click **Yes**. This saves the data source name.

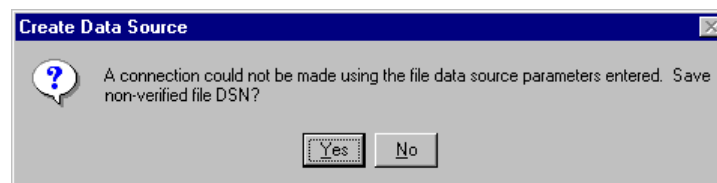


Figure 83. Create new OneWorld Data Source connection warning

- f. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except for the Data source name.
- g. If this ODBC Data Source was already defined, you can select **Cancel** and return to step 1 with the next OneWorld data source. If you prefer, you can select **Finish** and continue to redefine the ODBC Data Source.
- h. Click the **Server** tab.
- i. Enter the corresponding library name for this data source for Default Libraries, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
- j. Click the **Package(s)** tab.
- k. Enter the corresponding library name for this data source for Default package library, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
- l. Click the **Performance** tab.
- m. Select **Enable Lazy Close Support**. Select **Enable Data Compression**. The information should appear the same as shown in Figure 69 on page 199.
- n. Click the **Translation** tab.

- o. Select either **Do Not Translate CCSID 65535** or **Translate CCSID 65535**, corresponding to what is required for this data source, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
 - p. Click the **Format** tab.
 - q. Select **System naming convention (*SYS)** for Naming convention. The information should appear the same as the example in Figure 69 on page 199.
 - r. Click **Apply**.
 - s. Click **Yes** in reply to “Data source already exists Overwrite?”.
 - t. Click **OK**.
 - u. If the Create New Data Source form appears, repeat this series of steps for configuring ODBC Data Sources for the next data source.
4. When you are finished configuring the ODBC Data Sources, the Data Source Workbench form re-appears. The Detail Status now shows 60 for each of the OneWorld data sources.
 5. Click **Next**.

Configuring your environments

The Environment Workbench form now appears. Verify that each of your configured OneWorld environments is listed. The Environment Status should be 30 (with a Status Description of Validated). The Data Load Status should be 20 (Status Description of Finalized) for each environment except DEPB733 (B733 Deployment Server).

Select **Configure** on the Form menu to run the Installation Workbench.

Note

It is possible that the Windows NT Task Manager function may display a status of “not responding” for OneWorld during the execution of the Installation Workbench. This is a normal condition and may be ignored.

Verifying Installation Workbench

An Environment Database Creation report (R98403) is generated during the execution of the Installation Workbench for each environment defined in the plan. Review these reports for any instances of failure messages.

Note: In this report, the *no action* statement is acceptable for co-existence environments.

Chapter 7 of the *OneWorld Installation Reference Guide (All Systems)* provides additional details of acceptable errors.

When the Environment Workbench has completed, the Environment Status of each environment is 60 with a Status Description of Installed.

If your R98403 report shows errors, you need to review and resolve them before continuing. What needs to be done to resolve the problem depends on the type of error.

However, you should examine your ODBC Data Sources to ensure that they are correct, because this is a common error. To examine and correct any errors in one or more ODBC Data Sources, complete the following steps:

1. Click **Start->Settings->Control Panel**.
2. Double-click **ODBC Data Sources (32 bit)**.
3. Click the **System DSN** tab.
4. Repeat the following steps for each ODBC Data Source that you want to examine:
 - a. Double-click the name of the ODBC Data Source that you want to examine.
 - b. The Client Access Express ODBC Setup (32-bit) form appears. The information should appear the same as shown in Figure 66 on page 198, except for the Data source name.
 - c. Click the **Server** tab.
 - d. If necessary, correct the corresponding library name for this data source in Default Libraries, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 67 on page 198.
 - e. Click the **Package(s)** tab.
 - f. If necessary, correct the corresponding library name for this data source in Default package library, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 68 on page 199.
 - g. Click the **Performance** tab.
 - h. Ensure that **Enable Lazy Close Support** and **Enable Data Compression** are selected. The information should appear the same as shown in Figure 69 on page 199.
 - i. Click the **Translation** tab.
 - j. Ensure that either **Do Not Translate CCSID 65535** or **Translate CCSID 65535** are selected, corresponding to what is required for this data source, based on Table 16 on page 194. Otherwise, the information should appear the same as shown in Figure 70 on page 200.
 - k. Click the **Format** tab.
 - l. Ensure that **System naming convention (*SYS)** for Naming convention is selected. The information should appear as shown in Figure 69 on page 199.
 - m. Click **Apply**.
 - n. Click **Yes** in reply to “Data source already exists Overwrite?”.
 - o. Click **OK**.
 - p. If you receive an ODBC Failure message, click **Cancel**, because the ODBC Data Source may already exist.
5. When you have completed checking and correcting your ODBC Data Sources, click **OK** on the ODBC Data Source Administrator form.
6. If it was necessary for you to correct an ODBC Data Source, go back into your custom installation plan and change the Data Load Status for the failed environment(s) to 30. Once the plan is reset, you repeat configuring your installation plan in the Installation Workbench. This reads the files that were

already created or populated and the configuration is just run for the tables that need to be fixed. Complete the following steps:

1. Log on to J.D. Edwards OneWorld on the Deployment Server as user `JDE` and the planner environment `JDEPLAN`.
2. If necessary, type `GH961` on Fastpath and press Enter to display the System Installation Tools Menu.
3. Double-click **Custom Installation Plan**.
4. Click **Find** on the Work with Installation Plan form.
5. Click **Expand** on the Work with Installation Plan form.
6. Click **Environments** to highlight it, and then click **Review**.
7. On the Environment Planner display, click the environment on which you want to change Data Load Status. This should highlight that environment line.
8. Click **Data Load**.
9. On the Data Load Parameters display, change the Plan detail status to a **30 (Validated)**.
10. Click **OK**.
11. Click **Save** on the Environment Planner form.
12. Click **Cancel** on the Environment Planner form.
13. Click **Cancel** on the Package Planner form.
14. Click **Close** on the Work with Installation Planner form.
15. Double-click **Installation Workbench**.
16. Double-click your plan name.
17. Click **Next** on the Location Workbench form.
18. Click **Next** on the Data Source Workbench form.
19. On the Environment Workload form, click the environment on which you want to run the configuration. This should highlight that environment line.
20. Click **Configure**.

This should start the run through R98403 again. When finished, the Data Load Status should now show *60 (Installed)*.

Configuring your machines

To configure your machines, perform the following steps:

1. Click **Next** on the Environment Workbench form.
The Machine Workbench form appears.
2. Click **Configure** on the Form menu.
When OneWorld configures your servers, the Detail Status becomes 60 and the Status Description becomes Installed.
3. Click **Next**.

Configuring your packages

To configure your packages, perform following steps:

1. The Control Table Workbench form appears, showing a list of your packages.
2. Click **Configure** on the Form menu.

When OneWorld configures your packages, the Detail Status becomes 60 and the Status Description becomes Installed.

3. Click **Next**.

Finishing Installation Workbench

To finish the Installation Workbench, perform following steps:

1. A message like the one shown in Figure 84 appears, indicating the successful completion of the Installation Workbench.
2. Click **Finish** on the Form menu.

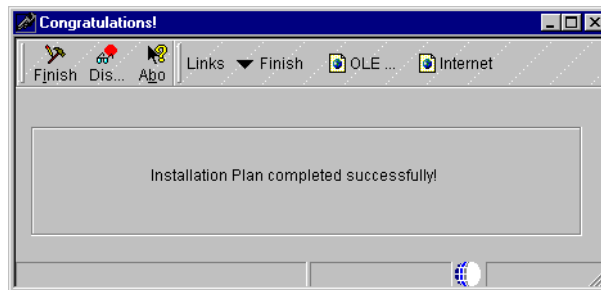


Figure 84. Installation Plan completion message

3. The Installation Workbench form re-appears, showing your package with an updated Plan Status of 60 and Status Description of Installed (Figure 85).

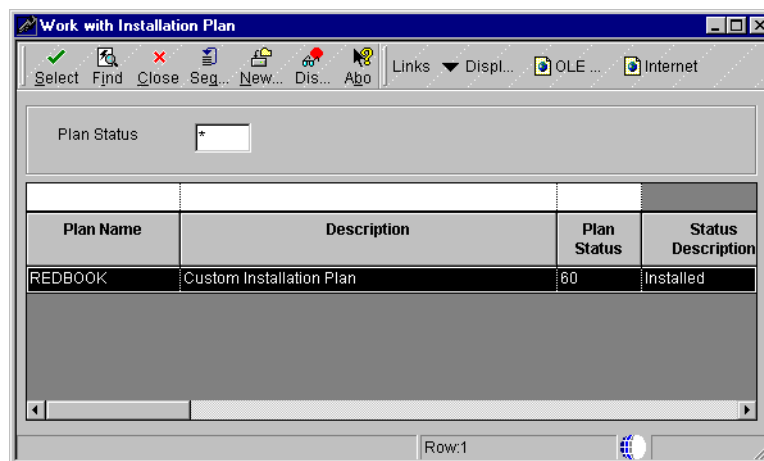


Figure 85. Finished Installation Plan

4. Click **Close**.

9.1.4 Enterprise Server installation

During this phase, the AS/400 Enterprise Server is configured and OneWorld objects are created along with associated IFS object links. Many of these

installation tasks are also required on any other server that you plan to include in your client/server environment, which will process OneWorld logic.

At Release B73.3, the majority of OneWorld objects are installed onto the Enterprise Server directly from the installation CD-ROMs. This reduces the volume of network traffic generated during the installation phase in comparison to earlier releases and subsequently reduces installation time. Objects containing OneWorld configuration information are still transferred from the Deployment Server to the Enterprise Server using FTP.

The host installation process loads the OneWorld services programs and the appropriate Path Code libraries. Each Path Code on the AS/400 server is comprised of specification files located in the integrated file system (for example /PRODB733/SPECFILE/*.*) along with business function programs located in a Path Code library, for example PRODB733. OneWorld services programs are restored into the B7332SYS library.

Before the AS/400 Enterprise Server installation phase is started, several configuration tasks must be performed on the AS/400 system and the Deployment Server. These are described in the following sections.

9.1.4.1 Setting the AS/400 system values

A number of AS/400 system values must be verified on the AS/400 server and amended as necessary. This can be done by typing the Work with System Values (`WRKSYSVAL`) command on an AS/400 command line followed by Enter. Then enter option 2 to change an individual system value. The values are:

- **QSECURITY:** J.D. Edwards recommends that you do not specify an OS/400 security level of 50 on a system running OneWorld, because the application will not function correctly. Therefore, if the QSECURITY system value is currently set to 50, it should be lowered to 40. An IPL is required for this change to take affect.

Note

The security level of your system should be ascertained during the initial configuration planning. If it is found to be set at 50, a review of your system security policy should be performed to verify that lowering it to 40 would not compromise the security requirements of your system.

- **QJOBMSGQFL:** This system value dictates what action is taken when a job's message queue becomes full. We recommend that you set this value to *WRAP or *PRTWRAP to ensure that the batch jobs submitted do not fail due to insufficient message queue space.

Note: This setting must be retained permanently.

- **QALWOBJRST:** This value must be set to *ALL for the duration of the installation process to allow programs adopting QSECOFR authority to be restored onto the system. This system value can be reverted to its original setting after the installation has completed successfully.
- **QQRYPDEGREE:** The QQRYPDEGREE system value controls the system-wide usage of parallel processing. Set this system value to *OPTIMIZE. This divides many database functions into separate tasks for simultaneous processing by one or more processors.

Note: The DB2 Symmetric Multiprocessing for AS/400 option of OS/400 is a pre-requisite for the activation of SMP.

For more detailed information regarding SMP, refer to Chapter 21, “Performance management” on page 561.

9.1.4.2 Creating OneWorld user profiles

The *OneWorld Installation Guide (AS/400 Systems)* states that an AS/400 user profile with *JOBCTL authority should be created at this time for each OneWorld user that will access the OneWorld AS/400 server. Although this needs to be done, it can wait until the completion of the Enterprise Server OneWorld installation. A user profile ONEWORLD is created during the Enterprise Server installation process and can be copied to these new OneWorld user profiles if done after the completion of the Enterprise Server installation.

9.1.4.3 Starting TCP/IP on the AS/400 system

Once TCP/IP is correctly configured on the AS/400 system, it is necessary to start it. You can do this using the Start TCP/IP (STRTCP) command. The required TCP/IP servers should automatically start as a result of the STRTCP command. If they do not, you can use the Start Host Servers (STRHOSTSVR) command to do this. For more information, refer to Chapter 7, “Work management” on page 109.

These tasks need to be accomplished each time your AS/400 system is started (IPLed). You can either do this manually, write a Control Language (CL) program, or modify the AS/400 Startup Program (as specified in system value QSTRUPPGM).

9.1.4.4 Amending the AS/400 startup program

If you want the OneWorld services to be automatically started at IPL time, the AS/400 system startup program must be amended to include the STRTCP *ALL and STRNET commands.

Note: In V4R3 and later releases, the STRHOSTSVR *ALL command is integrated into the STRTCP command.

We recommend that you modify the AS/400 startup program as follows:

1. Use the Display System Value (DSPSYSVAL) command for system value QSTRUPPGM to determine the name of the startup program for your AS/400 system:

```
DSPSYSVAL QSTRUPPGM
```

2. Use the Retrieve Control Language Source (RTVCLSRC) command to obtain the current source code for your AS/400 startup program:

```
RTVCLSRC PGM(QGPL/QSTRUP) SRCFILE(QGPL/QCLSRC)
```

3. Use the Edit File (EDTF) command to modify your AS/400 startup program by adding the STRTCP, STRHOSTSVR and STRNET commands. Your resulting startup program source should include Control Language statements similar to those shown in Figure 86 on page 224.

```

PGM
DCL VAR(&STRWIRS) TYPE(*CHAR) LEN(1)
DCL VAR(&CTLSBSD) TYPE(*CHAR) LEN(20)
DCL VAR(&CPYR) TYPE(*CHAR) LEN(90) VALUE('5716-SS1 (C) COPYRIGHT-
IBM CORP 1980, 1996. LICENSED MATERIAL - PROGRAM PROPERTY OF IBM')
QSYS/STRSBS SBSDB(QSPL)
MONMSG MSGID(CPF0000)
QSYS/STRSBS SBSDB(QSERVER)
MONMSG MSGID(CPF0000)
QSYS/RLSJOBQ JOBQ(QGPL/QS36MRT)
MONMSG MSGID(CPF0000)
QSYS/RLSJOBQ JOBQ(QGPL/QS36EVOKE)
MONMSG MSGID(CPF0000)
QSYS/STRCLNUP
MONMSG MSGID(CPF0000)
QSYS/RTVSYSVAL SYSVAL(QCTLSBSD) RTNVAR(&CTLSBSD)
IF COND((&CTLSBSD *NE 'QCTL QSYS ')*AND (&CTLSBSD *NE-
'QCTL QGPL ')) THEN(GOTO CMDLBL(DONE))
QSYS/STRSBS SBSDB(QINTER)
MONMSG MSGID(CPF0000)
QSYS/STRSBS SBSDB(QBATCH)
MONMSG MSGID(CPF0000)
QSYS/STRSBS SBSDB(QCMN)
MONMSG MSGID(CPF0000)
QSYS/STRSBS SBSDB(QSNADS)
MONMSG MSGID(CPF0000)
STRICP
MONMSG MSGID(CPF0000)
STRHOSTSVR SERVER(*ALL)
MONMSG MSGID(CPF0000)
ADDLIBLE JDEOW
STRNET
MONMSG MSGID(CPF0000)
DONE:
QSYS/RTVSYSVAL SYSVAL(QSTRPRIWTR) RTNVAR(&STRWIRS)
IF COND(&STRWIRS = '0') THEN(GOTO CMDLBL(NOWTRS))
CALL PGM(QSYS/QWCSTRS)
MONMSG MSGID(CPF0000)
NOWTRS:
RETURN
CHGVAR VAR(&CPYR) VALUE(&CPYR)
ENDPGM

```

Figure 86. AS/400 startup program source

4. Recompile your AS/400 startup program using the Create Control Language Program (CRTCLPGM) command:

```

CRTCLPGM PGM(QGPL/QSTRUP) SRCFILE(QGPL/QCLSRC) SRCMBR(QSTRUP) TEXT('AS/400
Startup Program')

```

9.1.4.5 Changing AS/400 host requirement for Client Access Express

Subsystem QSERVER runs the host server job for the Client Access Express file serving and database serving functions. To ensure better performance of Client Access Express, perform following steps:

1. Ensure that the automatic performance adjuster has been turned on.

Type `WRKSYSVAL QPFRADJ` on an AS/400 command line, and press Enter. Type option 5 to display the value. If the value of QPFRADJ is either 2 or 3, no changes are required. Otherwise, enter option 2 to change this system value to either 2 or 3.

2. Put the QSERVER subsystem in its own memory pool.
 - a. Using the Work with Shared Storage Pools (`WRKSHRPOOL`) command, determine a shared pool which has not been defined, yet. Select an unused shared pool (for example, `*SHRPOOL6`).
 - b. Assign some memory (for example, 50 MB) by typing this number in the Defined Size (M) field. For each active OneWorld user, allocate approximately 15 MB of memory.
 - c. Assign an activity level (for example, 10) in the Max Active field. Press Enter. The amount of memory assigned and the setting for the activity level is not critical since the performance adjuster has been turned on, so adjustments will be made by the AS/400 system as required.
 - d. Change the QSERVER subsystem description to use the shared pool that you just defined. For example, on an AS/400 command line, type the following command, and press Enter:

```
CHGSBSD SBSD(QSERVER) POOLS(1 *SHRPOOL6)
```

All new jobs started in the QSERVER subsystem will now use the new memory pool.

3. Change the QPWFSEVER class (used by Client Access Express jobs) Eligible to Purge setting to *YES.

This should only be done if there is limited memory on your AS/400 system. IBM ships QPWFSEVER with Eligible to Purge set to *NO. The maximum CPU time and Maximum temporary storage options should already be set to the recommended *NOMAX. However, if you need to make these changes, type on an AS/400 command line:

```
CHGCLS CLS(QPWFSEVER) PURGE(*YES) CPUTIME(*NOMAX) MAXTMPSTG(*NOMAX)
```

Press Enter.

For more information on these activities, refer to Chapter 7, “Work management” on page 109, and Chapter 21, “Performance management” on page 561.

Setting the number of QSERVER prestart jobs

The number of prestart jobs for program QZDASOINIT depends on the number of concurrent and active ODBC user connections. If you do not have enough prestart jobs, your system may be slow to make connections. Each OneWorld user typically uses five to eight ODBC connections. Each ODBC connection uses one QZDASOINIT program. Therefore, the initial value for QZDASOINIT should be set to approximately five times the number of expected concurrent connected users. The programs QZSCSRVS and QZSOSIGN are used for making the connection. Therefore the initial value for QZSCSRVS and QZSOSIGN should be the number of expected concurrent connected users.

As the determination of the correct number to specify involves a degree of approximation, we recommend that you review this setting at a time when a typical number of users are accessing the system.

It is important to keep in mind that the higher the number of prestart jobs specified, the greater the performance impact is on the system at IPL time or QSERVER subsystem startup time.

For in-depth information regarding ODBC performance, refer to 21.2.1.1, “Client Access Express ODBC setup for performance” on page 565.

Complete the following steps to make these changes for QZDASOINIT. Repeat the same steps, while changing the program name, to make the changes for QZSCSRVS and QZSOSIGN:

1. To see the current settings, type the following command, and press Enter:

```
DSPSBSD QSERVER
```

Enter option 10 to view the Prestart job entries. Enter option 5 for program QIWS/QZDASOINIT to display the details to determine the current settings.

2. To change the current settings, type on an AS/400 command line:

```
CHGPJE SBSD(QSYS/QSERVER) PGM(QIWS/QZDASOINIT)
```

Press F4 to prompt. Make the appropriate changes, and press Enter.

For the above changes to take effect, the QSERVER subsystem needs to be restarted. This can be accomplished by using the following steps:

1. To end the QSERVER subsystem, on an AS/400 command line, type the following command and press Enter:

```
ENDSBS QSERVER *IMMED
```

2. To restart the QSERVER subsystem, type the following command on an AS/400 command line and press Enter:

```
STRSBS QSERVER
```

3. To restart TCP/IP, type the following command on an AS/400 command line and press Enter:

```
STRTCP
```

4. To restart the required AS/400 host server programs, type the following command on an AS/400 command line and press Enter:

```
STRHOSTSVR
```

9.1.4.6 Modifying the Enterprise Server jde.ini file

We recommend that you make the following modifications to the Enterprise Server's jde.ini file on the Deployment Server. This file is located in the \JDEdwardsOneWorld\B733\hosts\as400\system name\ directory, where *system name* is the name of your AS/400 system.

- **Security:** Amend the DefaultEnvironment statement in the SECURITY section of the jde.ini file to specify a valid environment value (for example, PRD733 as in Figure 87). Deactivate the SecurityServer statement by adding a semicolon (;) in front of the statement for the duration of all installation and configuration processes (Figure 87). Activate it again, when you are certain that the installation is successfully completed.

```
jde - Notepad
File Edit Search Help
CRTDBPGM2=DBDRU_CH DBDRU_CP DBDRU_DT DBDRU_RQ DBDRUSQL DBMONCTL DBDR
CRTDBPGM3=DBSQL DBSQL_A DBSQL_D DBSQL_I DBSQL_M DBSQL_S DBSQL_U DBSQ
CRTDBPGM4=BNDSPVPGM(JDEKRNL JDELIB JDEIPC QBFCPRCD) ACTGRP(%S) OPTI
CRTDBPGM5=*DUPUAR) AUT(*ALL) TGTRLS(U4R3M0)
PrintUBEJoblog=FALSE
PrintUBEJoblogOnError=FALSE

[UBE]
UBEDebugLevel=0

[SECURITY]
DataSource=System - B733 - DNT
User=JDE
Password=JDE
DefaultEnvironment=PRD733
;SecurityServer=DENAS2
ServerPswdFile=TRUE
History=0

[LOCK MANAGER]
Server=
AvailableService=NONE
```

Figure 87. OneWorld Enterprise Server jde.ini file

- **Remove unwanted path codes:** Any statements in the jde.ini file relating to Path Codes that were not installed must be altered to reflect installed path code values. This avoids the production of error messages.
- **Update Server Name:** Any statements in the jde.ini file relating to the name of the server must be changed, if necessary, to a valid server name. The name of the server should not be left blank, even if the statement has been commented with a semicolon (;) in front of the statement. This avoids the production of error messages.

9.1.4.7 Checking the LIBRARY file

The LIBRARY file located in the \JDEdwardsOneWorld\B733\hosts\as400\system name\ directory should include all the Path Codes, which are to be installed. This file is used during the Enterprise Server Installation to determine which Path Codes are to be installed on the Enterprise Server. Refer to Figure 88 on page 228.

Note

The first entry in this file should *always* be B7332SYS.

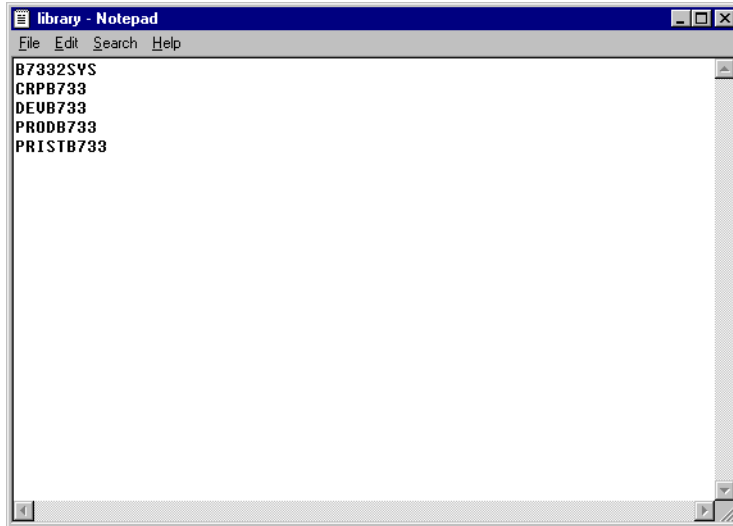


Figure 88. OneWorld AS/400 Library File

9.1.4.8 Installing OneWorld on the AS/400 Enterprise Server

Installation of the AS/400 Enterprise Server populates the libraries on the AS/400 system.

To install the Enterprise Server, perform following steps:

1. Sign on the AS/400 server as user QSECOFR.
2. Insert the OneWorld AS/400 Direct Enterprise Server Install 1 of 4 CD in the AS/400 CD-ROM drive.
3. On an AS/400 command line, type the following command and press Enter:

```
ADDLIBLE JDEOW
```
4. On an AS/400 command line, type the following command and press Enter:

```
GO JDEOW/A980WMNU
```
5. The menu shown in Figure 56 on page 179 appears. Enter option 4 and press Enter to install the OneWorld server libraries. This includes library B7332SYS, the path code (PRDB733, CRPB733, DEVB733, PRTB733) libraries as required and the appropriate integrated file system (IFS) directories. The AS/400 display shown in Figure 89 appears.


```

Install OneWorld Libraries (C98OWINST)

Type choices, press Enter.

Device Name . . . . . OPT01          Character value
Release Number . . . . . B7332
Deployment Server Name . . . . .
OneWorld Deployment Path . . . . . '?:\JDEwardsOneWorld\B733'
Deployment Server User ID . . . . . JDE          Character value
Deployment Server Password . . . . .          Character value
Change Messages to Break Mode . . . . . *YES      *YES *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 89. Install OneWorld Server Libraries on the AS/400 system

6. If necessary, change the Device name to that of your AS/400 CD-ROM drive (this is usually OPT01).
7. Type B7332 for Release Number.
8. Type the name of your Deployment Server.
9. Type the drive letter (for example, K) where OneWorld is installed on your Deployment Server, replacing the question mark in OneWorld Deployment Path.
10. Type JDE for the Deployment Server User ID.
11. Type the password for user ID JDE in Deployment Server Password.
12. Type *YES in Change Messages to Break Mode.
13. Press Enter.

The installation process submits the J98OW20 job to batch. Installation status messages may appear on the display.

14. When you are prompted for the second, third, and fourth CD, insert the CD in the AS/400 CD-ROM drive. When the AS/400 CD-ROM light stops flashing, type G in reply to the message, and press Enter.

The fourth CD may require a long time (may be hours depending on the number of OneWorld path codes that you are installing). While the installation is running, you can perform steps in 9.1.4.9, “Creating output queues” on page 230, and 9.1.4.10, “Creating AS/400 job queues” on page 230.

15. To monitor the status of the installation, on an AS/400 command line, type the following command and press Enter:

```
WRKSBMJOB
```

Find the job J98OW20. When the status of the job changes to OUTQ, the installation is complete. To confirm that the installation is successful, enter

option 8 to view the spooled files. Find the spooled file QPJOBLOG, and enter option 5 to display it.

9.1.4.9 Creating output queues

The OneWorld output queues are created at this stage by entering option 7 from the JDEOW/A98OWMNU menu and pressing Enter. The AS/400 display that appears is shown in Figure 90.

```
                                Create OneWorld Outqueue (C98OWOUT)

Type choices, press Enter.

Outqueue Name . . . . . Character value
Printer IP Address . . . . .
Remote Printer Queue . . . . . ' '
```

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 90. Creating a OneWorld Output Queue on the AS/400 system

Type your desired output queue Name, the appropriate Printer IP Address for your printer and, if necessary, Remote Printer Queue. Then, press Enter.

This creates a remote output queue with a TCP/IP connection. The value to be entered for the Remote printer queue parameter is specific to the make and model of printer to which the output queue is associated. This information should be available in the printer's handbook or alternatively from the manufacturer.

For more detailed information regarding the creation of OneWorld output queues, refer to Chapter 14, "Printing" on page 379.

9.1.4.10 Creating AS/400 job queues

Separate job queues can be created for each of the OneWorld modules. As an example, the following job queues can be created:

- JDEFIN
- JDEHRM
- JDEDST
- JDEMFG
- JDETECH

The actual names for these job queues should adhere to any naming conventions in place on your system. These are added to the QBATCH subsystem, or the alternative subsystem you chose for OneWorld batch processing.

Note

J.D. Edwards recommends that these job queues be single-threaded. That is to say the MAXACT parameter in a job queue entry should be set to "1". Adopting this setting removes the possibility of resource contention during the execution of concurrent UBEs within the same vertical OneWorld module.

Attention

At Release B73.3, the allocation of Enterprise Server job queues for UBE processing was changed to bring it in line with all other supported platforms. The software no longer relies solely on OS/400 work management concepts for the allocation of job queues. Instead, it now refers to a statement in the jde.ini file on the client system for details of which queue to submit UBEs into.

The OneWorld job queues are created at this stage by entering option 8 from the JDEOW/A98OWMNU menu and pressing Enter. The AS/400 display that appears is shown in Figure 91.

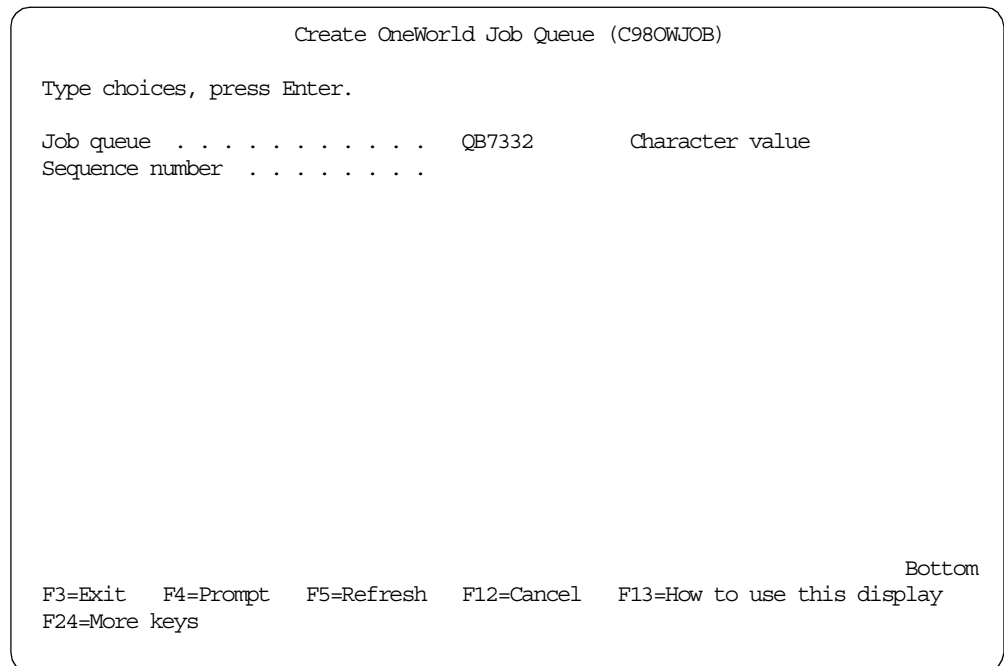


Figure 91. Create OneWorld Job Queue on the AS/400 system

Type your desired Job Queue Name and sequence number, and press Enter.

You need to determine beforehand the value to be entered for the Sequence number parameter. Enter option 6 on the JDEOW/A98OWMNU menu, and press

Enter. This shows you a list of sequence number already used for job queue so you can select another number.

9.1.4.11 Verifying SETUP.INF files on workstations

The *OneWorld Installation Guide (AS/400 Systems)* suggests that you modify the SETUP.INF file by adding a line `AllowDataCompression = 1` for each Client Access data source listed in the ODBCDataSource.inf file in the `\JDEdwardwsOneWorld\B733\client` directory. We found that this entry is automatically added during the installation process. You should verify that this line has been added to each Client Access data source in your ODBCDataSource.inf file, and if it is not, add it.

Note

The *OneWorld Installation Guide (AS/400 Systems)* states that you should set `LazyClose = 0` for each data source listed in the ODBC Datasource.inf file in the `\JDEdwardsOneWorld\B733\client` directory.

For performance reasons, set `LazyClose = 1` for each Client Access data source listed if you correctly set up your ODBC Data Sources in 9.1.3.3, "Creating a custom installation plan" on page 189.

9.1.4.12 Verifying the Enterprise Server installation

There are several tasks that you should perform to ascertain the success of the OneWorld Enterprise Server process. These tasks are described in the following sections.

Reviewing job logs and objects

When all of the batch jobs submitted by the host load process are completed, check the job logs for any errors.

Checking object counts

View the OneWorld libraries on the AS/400 server to ensure that they contain the correct number of objects. These numbers should be equal for all Path Codes. The number of IFS object links should also be checked, and again should be equal for all Path Codes. Table 17 provides a guideline for the approximate number of objects and IFS links for each OneWorld library.

Table 17. Number of objects and IFS links of OneWorld B733.2 libraries

Library	Approximate number of objects	Approximate number of IFS object links
DDB733	19	n/a
B7332SYS	115	2
CRPB733	29	28
DEVB733	29	28
PRISTB733	29	28
PRODB733	29	28

The number of objects and IFS object links may vary between releases. Contact J.D. Edwards for the exact numbers that are relevant to the release of OneWorld that you are installing.

Performing PORTTEST

PORTTEST is a J.D. Edwards-supplied application that performs environment initialization and verification. It may be run on the Enterprise Server as a post-installation verification process for all installed environments:

1. Sign on the AS/400 system as user `ONEWORLD`.
2. Enter option `1` for the version of OneWorld that you want to verify.
3. Type the following command, and press Enter:

```
GO JDEOW/A980WMNU
```

The OneWorld Installation Menu appears as shown in Figure 56 on page 179.

4. For each OneWorld environment that you want to verify, complete the following steps:
 - a. Enter option `12` to run the PORTTEST.
 - b. Type `JDE` for user ID, the password for JDE, and the environment that you want to test (for example, `CRP733`, `DEV733`, `PRD733`, `PRT733`, `TST733`). Press Enter.

Any failures during the processing of this application may suggest a problem with the configuration of the selected OneWorld environment. There are a number of conditions, which may cause PORTTEST to fail. These are listed in the Host Services Checklist at the end of Chapter 7 of the *OneWorld Installation Guide (AS/400 Systems)*.

As a general rule, before re-attempting PORTTEST after a failure, perform the following procedure on the Enterprise Server:

1. End OneWorld services with the command:

```
ENDNET
```

2. Clear inter-process communications:

```
CLRIPC
```

3. Sign off the AS/400 Enterprise Server.
4. Sign on as user `ONEWORLD`.
5. Start OneWorld services with the command:

```
STRNET
```

9.1.4.13 Starting the OneWorld AS/400 Enterprise Server

Start OneWorld services to ensure that all functions begin as expected. Complete the following steps:

1. Sign on the AS/400 system as user `ONEWORLD`.
2. Enter option `1` for the version of OneWorld that you want to start.
3. Type the following command, and press Enter:

```
GO JDEOW/A980WMNU
```

The OneWorld Installation Menu appears as shown in the example in Figure 56 on page 179.

4. Enter option 14 to end the JDE server.
5. Enter option 15 to clear the IPC.
6. Enter option 13 to start the JDE server.
7. Enter option 17 to display the active jobs.
8. Verify that there is an entry NETWORK with a program PGM-JDENET_N and status SELW.
9. Verify that there is an entry SENTINEL with a program PGM-MONITOR and status SIGW.

This starts the JDEB7332 subsystem and the associated SENTINEL, NETWORK, and JDENET autostrada jobs. These OneWorld services must be active before PORTTEST can be executed, if OneWorld Security Server or JDBNet processes have been configured for use.

9.1.4.14 Setting up a OneWorld printer

When the Enterprise Server installation is successfully completed, it is necessary to configure OneWorld printers. Perform the following steps:

1. Sign on the Deployment Server as user JDE in the deployment (DEP733) environment.
2. Type GH9013 in Fast Path, and press Enter to go to the OneWorld System Administration Menu.
3. Double-click **Printers**. The display shown in Figure 92 appears.

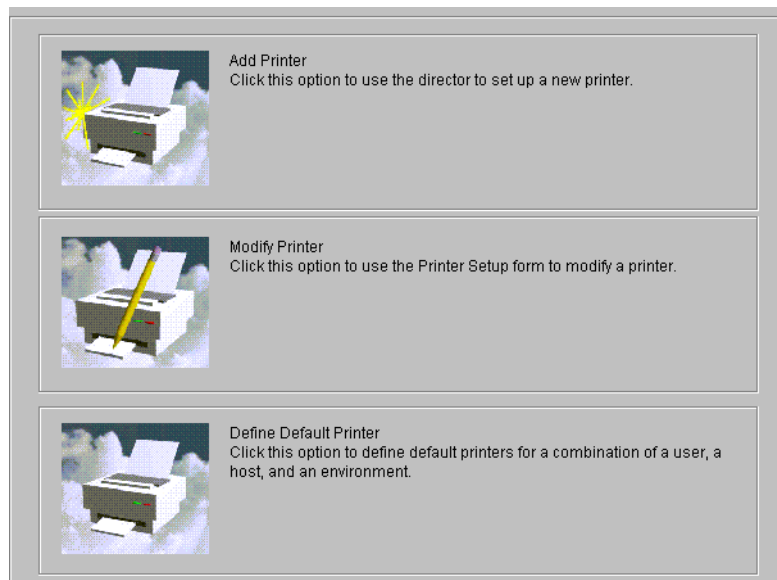


Figure 92. OneWorld printers menu

4. Click the icon to the left of Add Printer.
5. Click **OK** on the Printer Welcome form.
6. Type AS400 when prompted for Platform type. Click **Next**.
7. When prompted for printer information, type QUSRSYS for Library Name. Type the name of the output queue for output queue Name corresponding to the

output queue you created in 9.1.4.9, “Creating output queues” on page 230. Click **Next**. You are prompted for the location and model of the printer.

8. Use the visual assist to select your printer model (for example, Laser Printer, Line Printer). Type the location of the printer.
9. Double-click each row corresponding to supported paper types for this printer (you see a pink check mark next to the row).
10. Type 1 in Default Type for the paper type that is the default for this printer. This needs to be done even if there is only one supported paper type. Your display should look similar to the one in Figure 93.

Default Type	Paper Type	Printer Paper Width	Printer Paper Height
1	LETTER	8.50	11.00
	LEGAL	8.50	14.00
	A4	210.00	297.00

Figure 93. OneWorld printer model and location form (General tab)

11. Click the **Details** tab.
12. Click the appropriate Printer Definition Language for your printer. Your display should appear similar to the example in Figure 94.

Figure 94. OneWorld printer model and location form (Details tab)

13. Click **End**.

More information pertaining to the configuration of OneWorld printers is discussed in detail in 14.2, “OneWorld printing” on page 397.

9.1.5 Configuring and installing client workstations

During this phase, OneWorld user profiles are created. OneWorld objects and client configuration files are copied to client workstations.

The workstation installation and package deployment topic is very involved. It is covered in great detail in the *OneWorld Package Management Guide* and is also covered in more detail in Chapter 13, “OneWorld package management” on page 327.

The following sections provide a high-level overview of the related tasks required at this stage.

9.1.5.1 Creating user profiles and group profiles

OneWorld group profiles and user profiles must be created to allow environment and package deployment characteristics to be assigned prior to software deployment to client workstations. Each individual user can have access to their required environments. As an alternative, their valid environments can be determined from the group profile they are a member of, if any.

Note

Assignments made on individual profiles always override those made on the group.

For more detailed information regarding user profile creation, refer to 15.2, “OneWorld security” on page 420.

9.1.5.2 Installing OneWorld objects on client workstations

After OneWorld user profiles are created and assigned one or more deployment packages, they can be used to perform the workstation installation of OneWorld objects.

J.D. Edwards recommends that the initial installation of OneWorld objects on each client system is performed using the “Interactive” mode of installation. This allows the installer to select which deployment package to use for installation from those available to the user.

Running the setup.exe program located in the \client directory on the Deployment Server at a workstation initiates the OneWorld client installation process. This process must be repeated for each package (environment) that you want to install on each workstation, since only one package at a time is installed.

Before installing any packages on a new workstation, it is necessary to sign on the workstation as Administrator. Then you create a new user JDE to be used for installing the client workstation. This is the same process as described in 9.1.1.5, “Configuring the OneWorld Deployment Server” on page 171.

Sharing the B733 directory on the Deployment Server

We use the shared drive approach to installing the client workstation. To use a shared drive, complete the following steps on the Deployment Server before installing the first client workstation:

1. Using Windows NT Explorer, locate and right-click the **\JDEdwardsOneWorld\B733** directory.
2. Click **Properties**.
3. Select **Sharing** tab.
4. Click **Shared As**. Type **B733** for Share Name. Click **Maximum Allowed** for User Limit. Your display should look similar to the example in Figure 95.

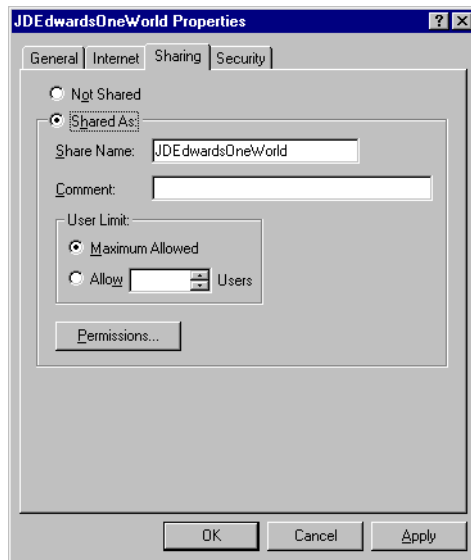


Figure 95. Deployment server directory sharing

5. Click **OK**.

Installing OneWorld in the interactive mode

The OneWorld client installation process may not run correctly if other applications are open. Therefore, ensure that all other applications (including virus checker, if applicable) are closed and disabled on the workstation.

Since we are using the shared drive approach to installing the client workstation, it is necessary to map a network drive on the workstation. Complete the following steps:

1. On the Windows Explorer display on the workstation, click **Tools**.
2. Click **Map Network Drive**.
3. Select a drive letter on your workstation that you want to use to refer to the Deployment Server OneWorld directory.
4. Type `\\server name\JDEdwardsOneWorld\B733` for Path Name, where *server name* is the name of your Deployment Server.
5. Select **Reconnect at Logon**. Your display should appear as shown in Figure 96 on page 238.

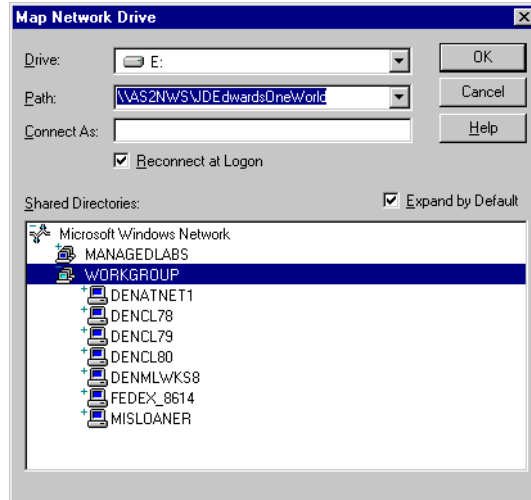


Figure 96. Map Network Drive on workstation

6. Click **OK**.
7. Expand **B733** on the mapped network drive.
8. Select **OneWorld Client Install** directory.
9. Double-click **InstallManager.exe**.
10. Click **Workstation Install** on the OneWorld Installation Manager form.
11. Click **Next** on the Client Workstation Setup form.
12. If you do not have the latest versions of Microsoft Internet Explorer or Adobe Acrobat Reader installed, the OneWorld Client Workstation Setup Third Party Application form appears, similar to the example in Figure 97. Complete the following steps:
 - a. Verify that the third-party application is selected. Click **Next**.
 - b. If you receive a warning during the installation of Acrobat 4.0 similar to Figure 98, click **OK**. Then, exit the installation, remove the old release of Acrobat, and restart the client OneWorld installation.
 - c. Click **Next** on the Acrobat Reader 4.0 Setup form.
 - d. Click **Accept** on the Software License Agreement form.
 - e. On the Choose Destination Location form, if you want to change the location where Acrobat Reader 4.0 will be installed, select **Browse** and select the desired location.
 - f. Click **Next**.
 - g. Click **OK** when the Thank you for choosing Acrobat Reader message is displayed.
13. Select the package that you want to install on the Client Workstation Setup Package Selection form. Click **Next**.

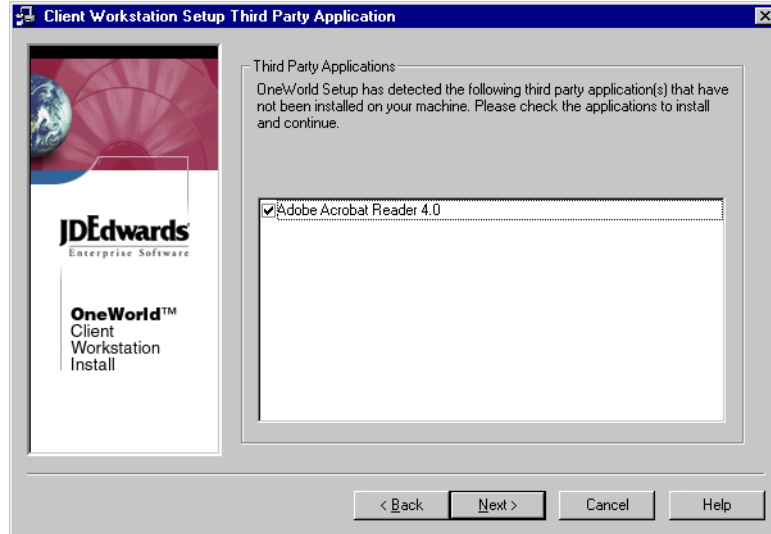


Figure 97. Client Workstation Setup Third Party Application installation

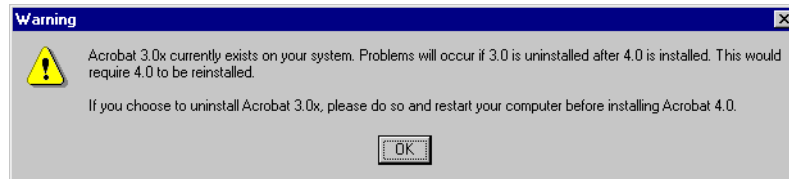


Figure 98. Acrobat 4.0 installation warning

14. If you plan to develop applications at this workstation, click **Development** on the Client Workstation Setup Type form. Otherwise, click **Production**.
15. To choose the disk drive on your client workstation where OneWorld is installed, click the ellipsis (...). The directory path *must* be \B7 on whichever disk drive you choose.
16. Click **Finish**.

When OneWorld is finished installing, you see a display like the example in Figure 99.

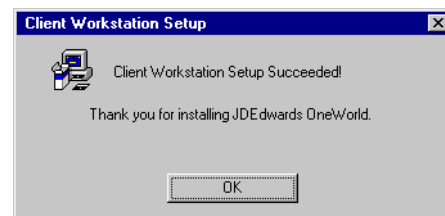


Figure 99. OneWorld Client Workstation Setup completion

Windows Terminal Server installation

OneWorld B73.3 includes support for Microsoft Windows NT 4.0 Terminal Server Edition (TSE), which allows a number of client workstations to connect to the Enterprise Server through a single client connection.

Currently Microsoft MDAC 2.1 is supported by Microsoft Windows NT 4.0 TSE. This is currently required if running SQL Server 7.0 with OneWorld B73.3.2. Therefore, complete the OneWorld client installation on the Windows Terminal Server.

9.1.6 Completing the installation

This section describes the tasks that you should perform following the main software installation procedures to ensure correct OneWorld functionality. These are necessary to remedy several known issues with Release B73.3.2 of OneWorld. It is anticipated that these will be resolved in future releases.

9.1.6.1 Amending Server Map Object Configuration Manager mappings

Following the installation of OneWorld on the Enterprise Server, amend the default setting for UBEs in the Server Map OCM to *not active*, only if necessary. When left *active* in previous releases of OneWorld, this has caused problems for UBEs, which call other UBEs. However, in Release B73.3.2, this should no longer be required.

Environment	Object Name	Object Type	Primary Data Source	User	Object Status	Data Source Mode	Second Source
CRP733	DEFAULT	UBE	ASM22	*PUBLIC	NA	P	
PRD733	DEFAULT	UBE	ASM22	*PUBLIC	NA	P	
WCRP733	DEFAULT	UBE	ASM23	*PUBLIC	NA	P	
WPRD733	DEFAULT	UBE	ASM23	*PUBLIC	NA	P	

Figure 100. Default Server Map OCM entries for UBEs

9.1.6.2 Amending B7332SYS library authority

There is a possibility that UBEs may fail immediately on submission due to authority problems. When a UBE is submitted, an attempt is made to delete a temporary password file owned by user profile ONEWORLD in the B7332SYS library. If the user profile submitting the UBE has insufficient authority to this file, the UBE fails.

To circumvent this problem, only if necessary, change the create authority for the B7332SYS library to *ALL. This allows any user with a minimum of *USE authority to the library to delete new objects created in the library. Issue the following command:

CHGLIB LIB (B7332SYS) CRTAUT (*ALL)

Since it is likely that the temporary password file is the only object created in this library after initial installation, this should not pose a great security exposure.

Library B7332SYS is created with CRTAUT value of *CHANGE, so this should be unnecessary.

9.1.6.3 Adding OCM mappings for Business Function B9861101

In some scenarios, users may not be able to view the output from previously submitted jobs. This problem can occur in any environment where the default OCM mapping for Business Functions is not set to LOCAL, typically environments set up for Windows Terminal Servers. This causes a failure in the Work With Servers application, P986110B.

To circumvent this, if necessary, create an OCM mapping for Business Function B9861101 in all environments for user *PUBLIC. The Primary Data Source for this OCM entry should be set to "LOCAL".

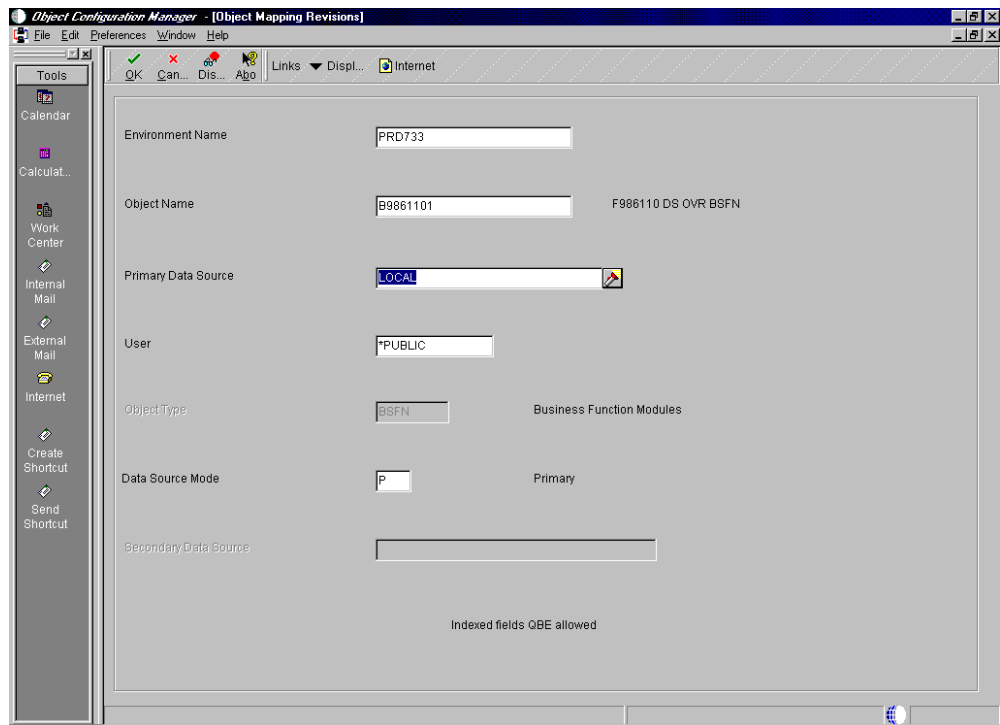


Figure 101. OCM Mapping Revisions display

9.1.6.4 Adding OCM mappings for F98306

No OCM mappings for the F98306 file are delivered in OneWorld B73.3. This condition causes the P98306 application to fail. To resolve this error, amend the OCM mapping entries for this file in all environments to map the file to the respective Central Objects data source.

This has been fixed in OneWorld Release B733.2, so you should not have to make this change manually.

9.1.6.5 Modifying media object queues

A media object queue must be defined to identify the pointer to the location where actual image files or OLE objects reside. For OLE objects, the name of the queue must be OLEQUE. The media object queues must reside on a Windows NT machine. If you want, you can use the Deployment Server (which is installed on an Integrated Netfinity Server) or another Integrated Netfinity Server, if you have multiple installed on your AS/400 system.

Media object queue paths are shipped with generic server names. To enable media object queues at your site, you must enter the Deployment Server name in the media object queue path. To do this, complete the following steps:

1. Sign on the client workstation with user `JDE` into the specific OneWorld environment where you want to make these changes.
2. Type `GH9011` in Fast Path, and press Enter to display the System Administration Tools menu.
3. Double-click **Media Object Queues**.
4. Click **Find** on the Work With Media Object Queues form.
5. Repeat the following steps for each media object queue:
 - a. Double-click a media object queue.
 - b. Type `\\server name\B733\queue name` for Path, where *server name* is the name of your Deployment Server and *queue name* is the name of the media object queue.
 - c. Click **OK**.

9.1.6.6 Copying Data Dictionary files to the Enterprise Server

The *OneWorld Installation Guide (AS/400 Systems)* describes the process of copying the Data Dictionary files back to the Enterprise Server. You don't have to do this, since this process was done by the installation process.

9.1.6.7 Changing *PUBLIC authority for JDE

To submit UBEs to execute on the AS/400 Enterprise Server, it is necessary that the OneWorld users have *CHANGE authority to the user profile JDE on the AS/400. Although *CHANGE authority can be granted for each individual AS/400 user profile, it is easier to provide it for *PUBLIC. If this violates your security policy, provide this authority just for the AS/400 user profiles corresponding to the OneWorld users. To grant *CHANGE authority to user profile JDE for *PUBLIC, complete the following steps:

1. Sign on to the Deployment Server as Administrator.
2. Click **Start->Programs->IBM AS/400 Client Access Express->AS/400 Operations Navigator**. This starts AS/400 Operations Navigator for accessing the AS/400 from the Deployment Server.
3. If necessary, expand **My AS/400 Connections** to see your AS/400 Enterprise Server.
4. Expand your AS/400 Enterprise Server.
5. When prompted for Signon to AS/400, type `QSECOFR` for AS/400 User ID and the corresponding password for Password. Click **OK**.
6. Expand **File Systems**.

7. Expand **Integrated File System**.
8. Click **QSYS.LIB**, to list the objects in this directory on the AS/400 system.
9. Find and right-click **JDE.USRPRF**.
10. Click **Permissions**.
11. Select **Change** for (Public).
12. Click **OK**.

9.1.6.8 Running the R4094A batch application

If you use OneWorld Inventory Management (system 41) or OneWorld Sales Order Management (system 42), you must run the R4094A batch application, version XJDE0001. This application can be run from any client either locally or on the server once for each environment where these systems will be used.

To run the R4094A application, complete the following process for each environment:

1. Sign on the workstation as user `JDE`.
2. Type `GH9111` in Fast Path and press Enter to display the Report Writer menu.
3. Double-click **Batch Versions**.
4. Type `R4094A` for Batch Application and click **Find**.
5. Select **XJDE0001**, and click **Submit**.

9.1.6.9 Building Word Search tables

The Word Search Build program populates the Word Search table (F91013) from information in the menu tables (F00821 and F0083). Word Search helps you locate a OneWorld menu if you do not know its name. The installation of these tables is optional.

To build the Word Search Tables, complete the following steps from any workstation in any environment, since this will complete the word search build for all environments:

1. Sign on as user `JDE` in the Pristine (`PRT733`) environment at any workstation where Pristine has been installed.
2. Type `GH9011` in Fast Path, and press Enter to display the System Administration Tools menu.
3. Double-click **Batch Versions**.
4. Type `R95012` for Batch Application, and click **Find**.
5. Select **XJDE0001**, and click **Submit**.
6. Type `1` for Option 1 on the Processing Options form to clear the Menu Word Search table before you rebuild it. Ensure that you have a backup copy of this table before clearing it.

9.2 OneWorld software maintenance

This section gives a high level overview of the processes involved in maintenance of J.D. Edwards OneWorld software when using the AS/400 system as an Enterprise Server. This information is intended to inform and give some insight. It

is not meant as a replacement to any of the appropriate J.D. Edwards guides which are provided when necessary.

9.2.1 Software Application Requests (SARs)

An SAR is generated when an issue is logged by the J.D. Edwards support line and proven to be an anomaly in the OneWorld software. Once an SAR is generated, other OneWorld administrators can indicate they are experiencing the same issue. SARs are prioritized based upon severity and affect on operation of software. An SAR can be used to address any appropriate J.D. Edwards software release. An SAR can be requested by contacting the J.D. Edwards Worldwide Customer Support response line.

For contact information on J.D. Edwards support, refer to Appendix E, "Support structure" on page 823.

9.2.2 Electronic Software Updates (ESUs)

With the release of the J.D. Edwards OneWorld B73.3.2 software, the paper fix process is eliminated. Paper fixes have been documents describing appropriate source code changes. These documents have been made available to customers as a response to particular SARs. Paper fixes can still be obtained from the J.D. Edwards Web sight at <http://www.jdedwards.com> for OneWorld Software Releases prior to B73.3.2.

With J.D. Edwards OneWorld software Release B73.3.2 and the advent of the ESUs, these source code changes are now automated. These ESUs can also be obtained from the J.D. Edwards Web site at: <http://www.jdedwards.com>

An ESU is a downloadable self-extracting executable file. When executing the self-extracting file, you are prompted for the directory in which the files are to be placed. You must verify that this directory structure is giving an accurate location for the PLANNER pathcode on the Deployment Server. ESUs will place implementation documentation in the directory in which the self-extraction placed the objects.

9.2.3 OneWorld service packs

Service packs are used to distribute system technical tool changes to a specific release of OneWorld independent of the cumulative update or upgrade process. Service packs are not the same as cumulative updates. They do not contain any application code changes or enhancements. An experienced OneWorld installer should install Service packs, adhering to the specific installation instructions shipped with the service pack.

Service packs are delivered on a CD and installed on the Deployment Server. Running the setup.exe program on the Deployment Server replaces the System directory on the Deployment Server with the new updated objects contained on the CD. OneWorld system code (library B7332SYS) on the AS/400 Logic Server is also updated using the service pack. Save files containing the updated System libraries are copied to the Deployment Server during the execution of the setup program.

The new System libraries are installed on the AS/400 server using the service pack installation instructions, which walk you through using FTP to copy the new

System save files over to the AS/400 server and restore them. The service pack completely replaces the System libraries on the AS/400 system. Because the business function service programs in the pathcode libraries (for example CFIN) have signatures based on service programs in the System libraries, the LNKBSFN command must be run to re-link the business functions to avoid signature violation errors.

9.2.4 Application Software Updates (ASUs)

An ASU is a J.D. Edwards OneWorld software update that is designed to meet a specific need. An ASU is comprised of specification objects in a Table Access Method (TAM) file format. These TAM files are used to modify or create objects within the OneWorld. J.D. Edwards creates ASUs for a variety of application needs such as yearly 1099 processing. An ASU is applied in a similar manner as a J.D. Edwards OneWorld Software Update. The primary difference between an ASU and an update is the scope of the operation. An ASU affects a specific application and its appropriate objects, where an update modifies a wide variety of objects and applications.

Each ASU is delivered with appropriate documentation for installation and implementation. The documentation is designed to address implementation of an ASU for all supported Enterprise Server environments.

9.2.5 Update process

The OneWorld update process is used to update your current OneWorld release to a new cumulative level at the current release, for example, B73.3.1 to B73.3.2. Cumulative updates contain fixes and enhancements to a specific release of OneWorld. They typically contain a large number of objects and include numerous application enhancements. The update process is detailed and complex. Therefore, it should be performed by an experienced OneWorld installer following the appropriate version of the *OneWorld Upgrade Guide (AS/400 Systems)*.

The standard update methodology has you update your CRP environment to the new cumulative release of OneWorld, while your production users continue to function as normal in the production environment. This scenario assumes that your CRP environment is functionally identical to your production environment. After CRP is updated and tested, you update your production environment by copying the updated CRP environment specifications over to production while preserving and converting your business data tables. If you are not running a live production environment, you can update all environments simultaneously.

The update process uses the Installation Planner and Workbench to define and execute the update just as in a new installation. The update process executes a series of merges that are used to merge the new specifications shipped in the updated package on the update CD into your existing release OneWorld specifications. Other table merges are performed to update the control tables, menus, Data Dictionary, and so on. The merge processes use a set of documented rules to merge the new specifications with any customizations you have made to the current release. The *OneWorld Upgrade Guide (AS/400 Systems)* details what rules are used when merging the new specifications with your existing specifications so that customizations can be preserved.

With Release B73.3.3, the process of going to any new cumulative level or major release level will be handled by the upgrade process.

9.2.6 Upgrade process

The OneWorld upgrade process is used to upgrade your current OneWorld release to the next major release of OneWorld, for example, B73.2.1 to B73.3.2. This is a detailed and complex process that should be performed by an experienced OneWorld Installer following the appropriate version of the *OneWorld Upgrade Guide (AS/400 Systems)*.

Note

The OneWorld upgrade process does not currently support release skipping, for example B73.1 to B733.2. J.D. Edwards plans to support this functionality in Release B73.3 and later.

The standard upgrade methodology asks you to upgrade your CRP environment to the new release of OneWorld, while your production users continue to function as normal in the Production environment. This scenario assumes that your CRP environment is functionally identical to your Production environment. After CRP is upgraded and tested, you cut over to production by copying the CRP applications to production while preserving and converting your business data tables.

If you are not running a live Production environment, you can upgrade all environments simultaneously. J.D. Edwards also supports a data only upgrade that does not preserve any customizations you have made to the OneWorld applications or any new applications you created. In essence, the data-only upgrade is a new installation, with the addition of table conversions to preserve your existing business data. This is the easiest way to upgrade if you have not made any customizations.

The upgrade process uses the Installation Planner and Installation Workbench applications to define and execute the upgrade just as in a new installation. The upgrade process executes a series of merges that are used to merge any changes you have made to the OneWorld software into the new release level specifications and tables. In other words, the upgrade process installs a new complete set of OneWorld Central Objects at the new release. Then, it merges in any changes the customer has made from their earlier release specifications. The *OneWorld Upgrade Guide (AS/400 Systems)* details what rules are used when merging customizations with the new OneWorld release.

Chapter 10. WorldSoftware and OneWorld

Both J.D. Edwards WorldSoftware and OneWorld application suites are available on the AS/400 system. WorldSoftware was designed for, and can only be implemented on, the AS/400 system from an AS/400 display or a display emulation session. On the other hand, OneWorld was designed for a client/server environment and can be implemented on the AS/400 system as well as other hardware platforms.

The focus of this chapter is to explain:

- WorldSoftware and OneWorld coexistence
- Migration from WorldSoftware to OneWorld

10.1 Coexistence of WorldSoftware and OneWorld

Coexistence means that both J.D. Edwards WorldSoftware and J.D. Edwards OneWorld software can use the same set of business data files at the same time without conflict. For example, one person can enter order information through OneWorld Sales Order Entry, while another person enters orders through WorldSoftware Sales Order Entry. Each person can see the other person's work as soon as it is entered.

10.1.1 Coexistence and cohabitation

While coexistence means that both OneWorld and WorldSoftware users share the same system *and* data, cohabitation means that they run on the same AS/400 system or within the same organization, but they are *not* sharing any data files or control files.

Because of hardware resources or lack of migration paths, customers may consider having both software packages installed, completely independent of each other. This allows for WorldSoftware to continue running the production, while clients configure and train in the OneWorld environment.

This chapter focuses on coexistence only.

10.1.2 Why install OneWorld

A customer who currently uses WorldSoftware may wonder why to consider installing OneWorld. Here are some reasons:

- **Multiple platform flexibility:** WorldSoftware only runs on the AS/400 system. While OneWorld also runs on the AS/400 system, it can also run on Windows NT servers (for example, IBM Netfinity servers or Integrated Netfinity Servers on the AS/400 system) and UNIX servers (for example, IBM RS/6000).
- **Runtime application partitioning:** With its Configurable Network Computing (CNC) architecture, OneWorld allows you to more easily expand to use multiple servers. These can be either all of the same platform (such as all AS/400 systems) or mixed platform (for example, AS/400 and Integrated Netfinity Servers). Because the objects are platform neutral, a single version of OneWorld applications can run on any of these platforms simplifying the addition of more logic servers as well as the incorporation of new platforms into your enterprise.

- **Graphical and Web-based computing:** The OneWorld Toolset can generate OneWorld applications for use with Java, Hypertext Markup Language (HTML), and Windows interface. Desktops, including network computers, that do not have OneWorld installed have access to OneWorld applications and data via a browser over the Internet or the corporate intranet.
- **Improved interoperability:** You can more easily interface with other applications and messaging systems with OneWorld than you can with WorldSoftware. OneWorld objects communicate with each other via messages. J.D. Edwards has documented the application program interfaces (APIs) for all of its major and master business functions. This facilitates real-time, bi-directional communications between OneWorld and other applications.
- **Additional functionality:** OneWorld Version B73.3 has many functions that you would have to upgrade your WorldSoftware to Version A8.1 to obtain. Such functions as Advanced Accounts Receivable, Quality Management, and Discrete Manufacturing are available to you by staying at Version A7.3 of WorldSoftware and adding these modules via OneWorld coexistence.

10.1.3 Structural differences between WorldSoftware and OneWorld

This section examines the differences in structure between WorldSoftware and OneWorld. While WorldSoftware is based on libraries and library lists, OneWorld uses data sources, Object Configuration Manager (OCM), and a path code to access objects.

WorldSoftware and library lists

WorldSoftware relies on library lists composed of AS/400 libraries. Only data that resides in a library that is part of a library list can be accessed. The library lists determine whether the user has access to the data sources and what data files can be accessed.

OneWorld data sources and Object Configuration Manager

OneWorld Object Configuration Manager maps data to data sources. It indicates where the data resides as well as on which machine the logic is run. The concept of data sources allows OneWorld to access data and run logic on multiple machines and platforms. If the data resides on an AS/400 system, the data source will point to a library on the AS/400 system. However, that library does not have to exist in the user's library list for it to be accessed. Object Configuration Manager allows data to be located in different areas and on different platforms. For example, one data source may point to an AS/400 library, while another may point to an SQL database on an AS/400 Integrated Netfinity Server with Windows NT. A third data source may point to an Oracle database on an IBM RS/6000.

The OneWorld path code

OneWorld relies on path codes to determine how it will work. OneWorld does not access the WorldSoftware JDFOBJ library. Instead, the specification files that compose the path code determine how OneWorld will function. Note that OneWorld does not access a source library either. Objects are developed directly in the development tool and do not require a source code library. The "source code" for most object types is held in the specification files.

10.1.4 Advantages and disadvantages of coexistence

The primary advantage of coexistence is that it allows companies to gradually migrate to the client/server and network centric-computing model. This makes the transition less risky than the “all or nothing” migration required by many software vendors. Other advantages of coexistence may include:

- It allows the system users to choose the system with which they are more comfortable. This can reduce training issues since people can continue their work as usual while they are training on the new set of software.
- Investment protection for existing technology, time, and training. Incorporating OneWorld does not require you to sacrifice your existing investments in the AS/400, WorldSoftware, WorldVision, or application training.
- The user can choose to run the software that performs best to accomplish the same task. For example, the user can enter the G/L journal entries using OneWorld and post those entries using WorldSoftware since the posting would run faster.
- The ability for the WorldSoftware customer to implement an e-commerce solution quickly. A client can install OneWorld on top of the existing WorldSoftware system. The client can continue to run production using the native AS/400 code, while enabling interoperable solutions that require OneWorld.

The potential disadvantages of WorldSoftware and OneWorld coexistence may include:

- Dual maintenance of control files that cannot be shared
- Increased amount of required DASD
- Short term solution: Application development is currently not synchronized between the two products
- More complexity in installation and instructions. See section on Logical File considerations
- Limited CNC configurations: WorldSoftware’s RPG programs cannot access data stored in other database platforms

10.1.5 Required software levels

For coexistence to be configured correctly, the release of OneWorld must match to a specific version of WorldSoftware. A WorldSoftware base release, cumulative PTF release and coexist PTF must all be installed before beginning your OneWorld install against WorldSoftware libraries. For example, to set up coexistence with the B73.3.2 Release of OneWorld, the A7.3 Version of WorldSoftware must be installed. Cumulative update 10, as well as coexistence update A73PC000X2, must also be applied. Table 18 shows some of the current information.

Table 18. WorldSoftware and OneWorld levels

OneWorld level	WorldSoftware base release	Minimum WorldSoftware PTF	Maximum WorldSoftware PTF	Coexist PTF level
B7.3.1	A7.3	A73PC00006	A73PC00008	A73PC000W1
B7.3.1.1	A7.3	A73PC00006	A73PC00008	A73PC000W1

OneWorld level	WorldSoftware base release	Minimum WorldSoftware PTF	Maximum WorldSoftware PTF	Coexist PTF level
B7.3.1.2	A7.3	A73PC00006	A73PC00008	A73PC000W2
B7.3.2	A7.3	A73PC00008	A73PC00008	A73PC000W3
B7.3.2.1	A7.3	A73PC00008	A73PC00008	A73PC000W4
B7.3.2.2	A7.3	A73PC00008	A73PC00008	A73PC000W5
B7.3.3	A7.3	A73PC00009	A73PC00010	A73PC000X0
B7.3.3.1	A7.3	A73PC00009	A73PC00010	A73PC000X1
B7.3.3.2	A7.3	A73PC00009	A73PC00010	A73PC000X2

Contact the OneWorld Technical Response Line or check the Knowledge Garden for the most recent requirements. Coexistent requirements include the recommended IBM PTFs. J.D. Edwards recommends customers stay current on all J.D. Edwards Service Packs and Cumulative releases.

10.1.6 User profile maintenance

The System - B733 data source contains some of the same tables that are used in WorldSoftware, such as the user and environments tables (F0092, F00921, F0093 and F0094). These tables are stored by WorldSoftware in the CLTSEC library. Because of the different functionality of the two products, customers cannot share user and environments tables between OneWorld and WorldSoftware. Another reason for this recommendation has to do with the way upgrades are performed in OneWorld. The customer must maintain separate sets of tables and perform dual maintenance for both sets of tables.

To create a user profile in WorldSoftware, follow these steps in the order they appear:

1. Create the JOBID for the new USER. Look at the JOBID for JDEPROD and the default correct parameters (WRKJOBID).
2. Create the USRPRF for the new USER. Look at the USRPRF for JDEPROD and the default parameters (WRKUSRPRF).
3. Sign on to JDE environment and create a JDE USRPRF. Look at JDEPROD for the default parameters. From Menu G94, select option 2.
4. Assign the new user to an appropriate Library List. From Menu G944, select option 6.

Note: These steps are necessary only for the users who need to access WorldSoftware. Refer to the *WorldSoftware Technical Foundations* manual for more information. For information on OneWorld user configuration, refer to Chapter 15, "Security" on page 415.

10.1.7 Installation issues

There is a OneWorld installation process designed to have OneWorld installed *after* WorldSoftware is installed (including the appropriate WorldSoftware PTF level and Coexistence PTF). The installation process assumes that business data tables are populated with data. Business data in WorldSoftware is stored in the CLTDTA library.

Non-business data are referred to as control or common tables. Some examples of control tables are comprised of Menus, UDCs, Data Dictionary, and Report Versions. In OneWorld, these tables are either stored in Control Table - Prod data source or in their own unique data source. In WorldSoftware, the majority of control tables are stored in CLTCOM library. The CLTCOM library is referred to as the common library because the control tables can be shared across multiple business data libraries.

In some circumstances, a customer may need to use a WorldSoftware process and want to run batch programs using native RPG code. If you need to install WorldSoftware after OneWorld is already running in production mode, please review 10.4, “Converting OneWorld environment to coexistence” on page 269.

10.1.8 Data Source setup issues

To set up coexistence environments in OneWorld, make sure to configure the following data sources during Installation Planner:

- System: B733
- Data Dictionary: B733
- Versions: B733
- AS/400 Common: B733
- Other types of data sources

10.1.8.1 Configuring the system: B733 data source

The System - B733 data source contains some of the same tables that are used in WorldSoftware such as the user and environments tables (F0092, F00921, F0093 and F0094). In addition, the System - B733 data source contains other tables, such as host configuration and other control tables, that do not exist in WorldSoftware. J.D. Edwards recommends that you maintain separate sets of security tables.

10.1.8.2 Configuring the Data Dictionary: B733 data source

Even though the OneWorld Data Dictionary files have the same file names as WorldSoftware, such as F9200, they have different formats. You can have the Data Dictionary files for WorldSoftware in your common library. Data Dictionary for OneWorld is stored in library DDB733. See 10.2.3.6, “The coexistence merges” on page 261.

10.1.8.3 Configuring the versions: B733 data source

WorldSoftware and OneWorld have unique ways of administering batch processes. WorldSoftware stores the Dream Writer in F98301 and associated files. These files should also be stored in the WorldSoftware Common library (CLTCOM). OneWorld stores the UBE specifications in the XXX733DNT library for every environment.

10.1.8.4 Configuring the AS/400 Common: B733 data source

With Release B7.3.3.2, only User Defined Codes and Next Numbers control tables can be shared. The files already exist in the CLTCOM library. The installation process must include creating this data source and pointing it back to the WorldSoftware library that already exists and contains all other control tables for WorldSoftware. Using a custom CNC configuration, the system administrator can map UDCs to CLTCOM or PRODCNTL. See Table 19 on page 254.

10.1.8.5 Configuring other types of data sources

When you set up environments using the OneWorld Installation Planner, the following data sources are set up for each environment:

- Central objects
- Versions
- Business Data
- Control Tables
- Object Librarian
- Do Not Translate (DNT)

The data source name depends on the environment you create. For example, if you create a test environment, you create a Business Data - TEST data source. The following list explains each data source type:

- **Central objects:** The Central objects data source references a DB2 UDB for AS/400 database. Previous to V4R3 or V4R4, central objects had to reside in a location that supports Oracle or SQL server databases. With the current OS/400 release, Central objects can reside on the AS/400 system using BLOB support.
- **Business Data:** Depending on the environment you create, the Business Data data source can be your test, production, or any other data library that currently resides on your AS/400 system (PRODDTA, CRPDTA, TESTDTA). When prompted for the Business Data data source during Installation Planner, modify this data source to call the WorldSoftware library on the AS/400 system. The tables in these libraries are populated with data already. You can also create a new Business Data library on your AS/400 system using the demo data from OneWorld. Note that an environment with Business Data created from OneWorld cannot be used to coexist. This new Business Data source may have data, depending on the type of OneWorld environment you create. If you build additional environments on your AS/400 system to use with OneWorld, you must create one or more of the following libraries before running Installation Workbench:
 - PRISTDTA
 - CRPDTA
 - TESTDTA
 - PRODDTA

Note: The PRISTDTA environment does not coexist with WorldSoftware. WorldSoftware delivered a JDFDATA library for pristine testing and upgrades.

- **Control Tables:** The Control Tables data sources for the CRP, test, and production environments normally contain the following files:
 - **F0002:** Next numbers
 - **F00021:** Next numbers by company fiscal
 - **F0004, F0004D, F0005, F0005D:** User-defined codes and language user defined codes
 - **F0082, F00821, F0083, F0084:** OneWorld menu files

For coexistence, you should use the next number and user defined code (UDC) tables from your existing WorldSoftware libraries. Typically, these files are located in the AS/400 Common library. If you do not share the next numbers tables, use partitioned sets of next numbers.

When you define your installation plan, configure the AS/400 Common data source to reflect your AS/400 Common library name. When your plan is created, the appropriate OCM mappings change to point to your WorldSoftware files.

The menu tables in the Control Tables data source have the same file names as in WorldSoftware, but different formats. Also, in OneWorld, the menu tables are normally replicated out to the clients. If you are currently a WorldSoftware user, you must have the menu files for WorldSoftware in your AS/400 Common library (CLTCOM)

- **Object Librarian Tables:** The Object Librarian Tables data source is limited to one data source per *system*, not per environment. Object Librarian contains information about all objects used in OneWorld. This central object repository is a starting point for application development and OneWorld object administration.

For coexistence, there is little consideration. WorldSoftware uses the Software Versions Repository, which is a program that also contains a record of every object. The two programs use different master tables, so you cannot share information between the two products.

- **Do Not Translate (DNT):** The DNT data sources access tables on the AS/400 system that contain BLOBs (Binary Large Objects). Each AS/400 library that contains DNT tables must have a DNT data source that is configured not to perform translation. The following standard DNT data sources are created automatically during Installation Planner:

- System: B733 - DNT
- Business Data: type - DNT
- Control Tables: type - DNT
type is PROD, CRP, TEST, or JDE.
- Versions: pathcode - DNT
pathcode is PRODB733, CRPB733, PRISTB733, or DEVB733.

The corresponding ODBC Data Sources, which you configure manually during Installation Workbench, as well as the Enterprise Server - Server Map data source, must be defined as “Do not translate CCSID 65535”.

10.1.9 WorldSoftware versus OneWorld tables and indexes

WorldSoftware and OneWorld access tables containing the business data. The indexes (or logical files) over the business data tables allow applications quicker and more precise access to the necessary information. OneWorld indexes can be distinguished from WorldSoftware by the naming convention. WorldSoftware indexes end in “LX” or “JX”, where X is a letter between A and Z. OneWorld indexes end in “_x”, where x is a number between 1 and 9.

For example, a WorldSoftware index built over the F0101 file may be named F0101LA. A OneWorld index built over the same file may be F0101_1. These logical files are created for WorldSoftware by DDS and RPG programs are compiled against these specific tables. OneWorld creates the tables using SQL and the WorldSoftware RPG programs cannot access these tables. RPG programs are compiled against a specific file format ID. WorldSoftware and OneWorld can share tables, but they must be created on the AS/400 system.

10.1.10 Control table comparison

Table 19 compares WorldSoftware and OneWorld control tables. It may help coexistence users to migrate from WorldSoftware to OneWorld environment. As of OneWorld Release B7.3.3.2, only the User Defined Code tables and Next Numbers tables can be shared between the two software releases.

Table 19. WorldSoftware and OneWorld control tables

Control table (OneWorld only)	WorldSoftware tables	WorldSoftware library	OneWorld tables	OneWorld library
Menus	F0082, F00821, F0083	CLTCOM	F0082, F00821, F0083	PRODCTL
User Defined Codes	F0004, F0004D, F0005, F0005D	CLTCOM	F0004, F0004D, F0005, F0005D	CLTCOM
Next Numbers	F0002, F00021	CLTCOM	F0002, F00021	CLTCOM
Data Dictionary	F9200, F9201, F9202, F9203, F9204, F9205, F9206, F9816, F98163	CLTCOM	F9200, F9201, F9202, F9203, F9204, F9205, F9206, F9816, F98163	DDB733
DREAM Writer or (Universal Batch Engine)	F98301, F98302, F98303, F9831, F98311, F98312, F81900, F81901, F81902	CLTCOM	F983051	PRD733DNT
User Security	F0092, F00921, F0093, F0094, F0095	CLTSEC	F0092, F00921, F0093, F0094, F0095	SYSB733
Vocabulary Overrides and Function Keys	F9220, F9601, F9601D, F9611, F9620, F9621	CLTCOM	F9220	?
Software Versions Repository or (Object Librarian)	F9801, F98012, F9802	CLTCOM	F9860, F9861, F9862, F9863, F9865	OLB733
Generic Text Files	F0016, F00161, F00162, F00163, F00164	CLTDTA	F0016, F00161, F00162, F00163, F00164	CLTDTA (must use triggers)

10.1.11 Data dictionary file synchronization

You must separately maintain the OneWorld Data Dictionary and the WorldSoftware Data Dictionary. This means that any changes you make to the OneWorld Data Dictionary you must also make in the WorldSoftware Data Dictionary, and vice versa. To help you maintain parity between the two data dictionaries, J.D. Edwards provides a batch process that compares both data dictionaries on the AS/400 system (see 10.1.11.1, "Synchronizing data dictionaries" on page 255). If you run this batch process, the following actions occur:

- If an item is in the WorldSoftware Data Dictionary but not in OneWorld, this process adds it to OneWorld.
- If an item is in OneWorld but not in WorldSoftware, the process prints this information on an exception report.
- The process replaces the glossary information, row headings, and column headings in OneWorld with that information from WorldSoftware.

- Item specifications that both WorldSoftware and OneWorld use, such as data item size, system code, or decimal places, are overwritten in OneWorld with information from WorldSoftware.

The two data dictionaries should not go out of sync if you are adding and changing items to both data dictionaries. However, for integrity reasons, you should run this report first in proof mode and, if necessary, in update mode to keep the data dictionaries alike.

10.1.11.1 Synchronizing data dictionaries

To synchronize coexistence data dictionaries, perform this process:

1. Sign onto the AS/400 WorldSoftware, and select the environment where the WorldSoftware Data Dictionary resides.
2. Select **DREAM Writer, Versions List**.
3. Enter `Form P99800`, and select **Version XJDE0001**.
4. In Processing Options, enter the library that contains the WorldSoftware Data Dictionary and the library that contains the OneWorld Data Dictionary. To start the update, enter `1` for update mode. To review the update before the process actually begins, enter `0` for proof mode.

The update process takes about 30 minutes. It produces reports with the following information:

- Data items added to OneWorld
- Items in OneWorld that are not in WorldSoftware
- Differences found

10.1.12 User Defined Code synchronization

The User Defined Code Tables should be updated from the WorldSoftware side and merged into the OneWorld side. Because WorldSoftware updates will be the base for what the UDC will do, OneWorld is simply reading that data from WorldSoftware. The R98403 UBE has these merges available for use as the software is implemented.

Also, if you are keeping UDC tables in the Local Access Databases of your users, you must merge the data into the `\b733\Pathcode\Package\packagename\data\jdeb7.mdb` Pathcodes Local Access Database. After this merge is done, the next step is to recompress the data part of the Full package you updated and then push that out to your clients using an Update package. If an item is in OneWorld but not in WorldSoftware, the process prints this information on an exception report.

10.1.13 Coexistence triggers

There are certain types of non-business data, generic text for example, that are held in a different format (and, therefore, a different file) between WorldSoftware and OneWorld. For these files to stay in synchronization, a coexistence trigger is required.

For detailed information on coexistence triggers and how to set them up, see the *OneWorld Application Coexistence Guide*.

10.1.14 Non-coexistence applications

The J. D. Edwards coexistence strategy has its limitations. Application development is not always in sync between the two software packages.

Note

Accounts Receivable and Human Resources are *not* coexistent. These applications cannot share tables because of table changes or specific application functionality. Please consult individual application support for more information.

What this means for a current WorldSoftware client wanting to move to OneWorld is some applications can only be run in WorldSoftware or OneWorld. For example, Accounts Receivable (A/R) needs to run a set of conversion programs to reformat their A/R data. Once this is done, they can access their A/R data through OneWorld only.

10.2 Moving from WorldSoftware to OneWorld

In this section, we introduce you to the concepts and requirements for moving from your AS/400 WorldSoftware environment to OneWorld, using the coexistence process. In the following section, we provide a step-by-step procedure for actually implementing the coexistence environment.

10.2.1 Coexistence scenarios

There are three different scenarios in which to consider coexistence of WorldSoftware and OneWorld:

- **New J.D. Edwards customers:** A new customer to J.D. Edwards may have one or more of the following requirements:
 - Does not need all of the functionality that OneWorld has to offer for all applications and is satisfied with the functionality of specific WorldSoftware applications.
 - Prefers the simpler interface of WorldSoftware for particular users and applications, particularly if moving from another legacy application on an AS/400 system.
 - Has an extensive investment in AS/400 hardware and fixed function terminals. This customer may prefer to continue to use as much existing hardware as possible and add full function workstations gradually, as required.

Coexistence can be very beneficial in these situations as WorldSoftware can be used where fixed function terminals are deployed, or where WorldSoftware functionality is sufficient. OneWorld can be deployed where the additional functionality is required.

In this situation, this customer installs WorldSoftware first and then installs OneWorld. This is J.D. Edwards' recommended approach to implementing a coexistence environment.

- **Existing WorldSoftware customers:** The existing WorldSoftware customer may consider OneWorld for one or more of the following reasons:

- Needs the additional functionality in OneWorld that cannot be found in WorldSoftware.
- Wants to migrate to a modern network-centric solution.
- Needs better integration with other systems such as data warehousing and e-commerce.
- Needs a more powerful user interface than is available with fixed function terminals.

This type of customer is looking for the improvements that OneWorld offers without the disruption and learning curve of a total replacement of the existing WorldSoftware solution. Coexistence offers this customer a gradual migration approach to the benefits available with OneWorld.

As in the previous scenario, this customer installs OneWorld on top of the existing WorldSoftware database. Clearly, this has to be planned carefully to minimize the unavailability of the system while OneWorld is being installed and to maintain the integrity of the existing WorldSoftware database.

- **Existing OneWorld customers:** There are various situations when an existing OneWorld customer needs to implement specific WorldSoftware applications. This customer requires “reverse” coexistence. Although it may be possible to accomplish this in specific situations, this can be very complex. There is also no formal, nor J.D. Edwards supported, solution to perform this reverse coexistence. Therefore careful planning, implementation, and testing is necessary if you need to pursue this option. For assistance with this reverse coexistence, review the process described in 10.4, “Converting OneWorld environment to coexistence” on page 269.

10.2.2 Preparing the AS/400 system for OneWorld

Moving to OneWorld in an AS/400 environment involves several different components and options. This section explains these requirements.

10.2.2.1 System requirements

The AS/400 requirements for OneWorld and WorldSoftware coexistence are:

- A RISC processor, since OneWorld does not run on the CISC processor.
- OS/400 V4R2 (or newer) operating system, which includes the necessary TCP/IP software, DB2 UDB for AS/400, and Client Access component necessary to run OneWorld.
- AS/400 ILE/C compiler (requires a separate license from OS/400).
- The most current cumulative program temporary fix (PTF) update tape for your release of OS/400 plus PTFs that are specific to supporting OneWorld. For more information on specific OS/400 PTFs required for OneWorld, review APAR II11677 for OS/400 V4R3M0 or APAR II11881 for OS/400 V4R4M0 at the Web site: <http://www.as400.ibm.com/service/bms/jde-support.htm>
- OneWorld is a client server model of computing. One additional requirement is to determine a client strategy. Heavy clients are one option (Windows 95, Windows 98, or Windows NT). A second option is to employ Windows Terminal Server, which usually requires an outboard multi-processor Windows NT system, such as the IBM Netfinity 7000 series.

10.2.2.2 AS/400 upgrade choices

As a WorldSoftware customer, you already have AS/400 hardware in place. Based on the age and capacity of your hardware, upgrading your AS/400 system may offer advantages when moving to OneWorld.

The AS/400 7xx series allows a great deal of flexibility in configuration to support operating environments that blend client server (OneWorld) and traditional interactive activity (WorldSoftware.) These machines can be configured to match anticipated blends of OneWorld and WorldSoftware usage. Since your WorldSoftware and OneWorld blend can differ based on business needs and timing, the configuration can be changed to match your needs.

10.2.2.3 Budget considerations

In addition to these technical requirements, there are other areas that you need to consider as you move to network computing. The following list presents some of the additional items that you need to consider:

- **Deployment Server and a database:** The cost of a Deployment Server can vary depending on how powerful the machine is that you choose. With V4R3 and V4R4 of OS/400, you can use DB2 UDB for AS/400 to store the specifications (Central objects) and do not need a database on the Deployment Server. In addition, the AS/400 Integrated Netfinity Server can be used as your Deployment Server, without the requirement for a standalone Windows NT server.
- **Hardware:** For your AS/400 system, if you do not already have a RISC processor, you will need to obtain one. This entails increasing the disk storage and doubling the memory to support just your current WorldSoftware implementation. Your AS/400 requirements for your new OneWorld implementation will need to be added to this. For more information on OneWorld sizing, refer to Chapter 8, “Pre-installation and planning” on page 139.
- **Network:** Depending on your current network infrastructure, the cost of providing a suitable TCP/IP network for OneWorld involves a number of factors, including the number of users, number of LAN segments, WAN connection methods, and so on.
- **Software:** Depending on whether you are currently on a user-based, suite-based, or model-based licensing program, there may be a charge for migration to OneWorld. You should consult your J.D. Edwards installation base representative or client manager for details about any applicable pricing migration fee.
- **Training:** J.D. Edwards’ OneWorld technical training curriculum has been designed to help technical professionals expand their knowledge. As with any new application suite, all individuals in your organization need to plan and complete applicable training in order to ensure a successful implementation.

10.2.3 Overview of a coexistence installation plan

Note

Before you begin the OneWorld installation, make sure that *all* system codes have been installed in the WorldSoftware installation. If they are not, you will end up with data tables created by OneWorld that are not readable by WorldSoftware.

The purpose of this section is to supplement the *OneWorld Installation Guide (AS/400 Systems)* with additional information regarding coexistence installation issues and procedures.

10.2.3.1 The coexistence plan

The main part of any OneWorld installation is the plan. The purpose of any OneWorld plan is to define where data resides in the installation. In a coexistence installation, it is critical that your data sources match up with your WorldSoftware installation's library structure. This means that before you begin, you need to have a clear understanding of what data resides in what library on the AS/400 system.

One of the first steps in any installation is the creation of the Data Dictionary. Restoring save files from the included host CD creates this. There is a menu that needs to be run from the CD that does this step fairly automatically. However, it is critical that this step is done before the install plan is done. For more information on the steps that you perform, refer to 9.1.1.8, "Installing libraries and Data Dictionary on the AS/400 system" on page 178.

10.2.3.2 Data sources

Each data source requires a specific setup. While the shared data sources use the same library as the WorldSoftware data, there are other OneWorld data sources that do not use the same library as the WorldSoftware data that has the same function, such as security. The actual data and other items, such as elements of the Common library, need to be identified.

When defining OneWorld data sources, the AS/400 database name, which is configured on the AS/400 system using the WRKRDBDIRE command, is usually the name of that AS/400 system. The local RDB name must be configured whether there is one RDB or multiple ones. The actual library name is the library that OneWorld uses to call for data.

The data source Advanced tab has several additional fields. These fields tend to be more generic. However it is critical that the correct entries are placed in these fields. These fields include:

- **Data Source Type:** For coexistence or non-coexistence, this must always be set to I for Client Access. Do *not* use the DB2 option for DB2 UDB for AS/400.
- **Object Owner ID:** This field is not used for AS/400 data sources.
- **Library Name:** This is the name of the AS/400 library where the data resides.
- **Library List Name:** This is the name of the AS/400 RDB that OneWorld and WorldSoftware use.

- **DLL Name:** This field must be JDBODBC.DLL for OneWorld to use ODBC to communicate with the AS/400 system.
- **Database Name:** This is the name of the ODBC Data Source called by OneWorld to communicate with the AS/400 system.
- **Server Name:** This is the name of the AS/400 server on which the data resides.
- **Platform:** This is the type of machine where the database resides (must be AS400).

After the data source has been defined, you need to configure the ODBC Data Source. To configure the ODBC Data Source, the first step is to choose the type of data source. The recommended type is a System Data Source.

After the type is chosen, choose the Client Access driver for ODBC. Then, complete the various tabs for each data source. These tabs must be correct or OneWorld will not be able to make a connection.

The data source name is the same as the Database Name in the OneWorld data source.

The Default libraries field lists the DB2 UDB for AS/400 library the data source will request when OneWorld talks to the ODBC driver.

The last note for data sources is the way the ODBC is setup to actually request data from the DB2 UDB for AS/400. System naming convention uses the Library name and File name as the full table name. Since WorldSoftware uses the system naming convention, a Coexistent OneWorld installation must also use the same naming convention. If you miss this setting, you will erase your WorldSoftware data during later stages of the installation.

J.D. Edwards recommends that you *do not unify* the following data sources with WordSoftware:

- System
- Data Dictionary
- Versions
- Object Librarian

These data sources are different enough that they need to remain separate from their WorldSoftware counterparts. The Data Dictionary Data source needs to be separate since the Items used by the two application suites are not unified at this time. The lack of unification in the formats of Windows NT and the AS/400 system have also led to a specific need on the AS/400 system for the DNT data sources.

10.2.3.3 DNT data sources

The DNT data source was created specifically for the AS/400 platform. It is used for OneWorld tables whose content must not be translated by Client Access drivers. This is accomplished by creating separate OneWorld data sources and mapping them onto IBM Client Access data sources that are set to not translate records. These data sources are labeled as “Do Not Translate” data sources, or DNT for short.

To create a OneWorld DNT data source, the changes are all done at the ODBC level. For *all other data sources*, you select **Translate CCSID 65535** on the

Translation tab. For *DNT data sources*, you must select **Do not translate CCSID 65535**. It is important to check that these data sources are in the installation plan and are also configured correctly in the ODBC manager.

10.2.3.4 Central objects

When you are configuring your data sources, you must also define the data sources for your Central objects. Prior to V4R4 of OS/400, Central objects needed to be placed either in Microsoft SQL Server or Oracle database. J.D. Edwards has developed a process that allows having Central objects in DB2 UDB for AS/400. This process can be used with either OS/400 V4R3 or V4R4. The resulting OneWorld Central objects data sources are also defined as Do Not Translate ODBC Data Sources on the AS/400 system.

For more detailed information on the definition of OneWorld Data Sources and ODBC Data Sources, refer to “OneWorld and ODBC Data Sources” on page 191.

10.2.3.5 Data load setup for environments

The data load for each environment is critical to the success of a coexistence install. The way the data load is configured determines what will happen to the existing WorldSoftware data. A wrong setting can destroy a significant portion of the WorldSoftware data. The correct setting for the data load is to run the data load for production data. This setting checks for existing tables and takes no action if they are already present. Should the table not be present, it simply creates the needed table with the minimum needed data.

The setting for data load is set to Load Production Data, for all environments except Pristine. For the Pristine environment, which does not share any data with WorldSoftware, you use a setting of Load Demo Data.

10.2.3.6 The coexistence merges

These conversions are designed to move UDC and Data Dictionary values from WorldSoftware to OneWorld. This is a merge of data specifically designed for a coexistence setup. The merges must be done to enable OneWorld to use data in the WorldSoftware installation. The UDC tables and Data Dictionary tables in WorldSoftware reside in a Library usually denoted as Common. The OneWorld tables (with the same file names as the WorldSoftware tables) reside in a different library, usually called control or CTL. The tables that reside in the CTL library include: F0004, F0005 (the UDC Tables), and F0082, F0083, F0084 (the menu tables). The Data Dictionary is normally in its own library in a OneWorld installation, but is kept in the Common Library for a WorldSoftware installation.

Remember it is critical that the Common data source be setup during the plan and point to the common library of the WorldSoftware installation, so that these merges may find the data in the WorldSoftware installation that needs to be moved to the OneWorld tables.

It is important to understand that WorldSoftware is also flexible and the control tables may easily reside in a separate library location on the AS/400 system. This means that the installer must be clear as to where the tables reside.

To ensure that these merges occur, ensure that you select Coexistent Merges in the Control Table Merges section of the Environments Advanced Parameters form.

10.2.3.7 Finishing the installation plan

Once these steps are done and setup, the next phase is to run the Workbench process. This step takes your various plan pieces and installs them to the OneWorld system database, making your plan live. The steps to a successful coexistence installation are not really all that different from any other installation. The location of the data and what data needs to be merged add complexity to this installation.

10.3 Converting WorldSoftware environment to coexistence

To convert a WorldSoftware AS/400 environment to a coexistence environment, you install OneWorld on an existing AS/400 system that has WorldSoftware already installed. Before beginning the installation of OneWorld, you must complete the following steps:

1. Ensure that you have installed V4R3 or V4R4 of OS/400.
2. Ensure that you have installed the most current OS/400 cumulative PTF tape.
3. Ensure that you install any additional OS/400 PTFs that are a prerequisite to installing OneWorld in a coexistence environment. If you are using V4R3 of OS/400, you must have PTF SF56894. If you are using V4R4 of OS/400, you must have PTF SF57165 and SF58404. If you have installed the most current OS/400 cumulative PTF tape, these PTFs should already be installed. Refer to Web site <http://www.as400.ibm.com/service/bms/jde-support.htm> for more current PTFs that may be required.
4. Ensure that your installed WorldSoftware is at Release A7.3 with Cumulative update 9 (A73PC00009) or higher.
5. Ensure that you install WorldSoftware Cumulative Update A73PC000X2, which is the coexistence cumulative update. To do this, you use the instructions in the *A7.3 PTF Install Workbook*, in conjunction with the following steps:
 - a. Before installing PTF A73PC000X2, ensure that no WorldSoftware user and no WorldSoftware job are accessing the data library that is in your upgrade plan.
 - b. Ensure that all Post Install jobs for the WorldSoftware updates have been applied. This includes working with merge reports, performing ASIs, creating new files and executing Special Application Jobs. Refer to *A7.3 PTF Install Workbook* for more information on Post Install jobs.
 - c. Move the generic text files to the production data library. The files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164LA, and F00165 usually reside in the WorldSoftware user library that contains the common files. We refer to this library by the J.D. Edwards standard name of CLTCOM. Your common library may be named differently.

If these files reside in the CLTCOM library, they should be moved to the WorldSoftware library that contains the production data files. We refer to this library by the J.D. Edwards standard name of CLTDTA. Your production data library may be named differently. Note that file F0016D does not need to be moved because it is replaced by file F00165.

After installing PTF A73PC000X2 but before beginning the Coexistence Special Jobs on the G97UX2 menu, move the control files F0016, F00161,

F00162, F00163, F00163LA, F00164, F00164LA, and F00165 one at a time from the CLTCOM library to the CLTDTA library. From an AS/400 command line, enter the following command:

```
MOV OBJ (CLTCOM/file) OBJTYPE(*FILE) TOLIB (CLTDTA)
```

Here, *file* is the control file to be moved. Press Enter.

- d. The Production files F01131, F01132, F4201, F4211LA, F4301, F4311LA, F4314, F4314H, F4802, and F4802H should only exist in the CLTDTA library. To determine which libraries currently contain these files, enter the following command on an AS/400 command line:

```
WRK OBJ (*ALL/file) OBJTYPE(*FILE)
```

Here, *file* is the production file that should only be in the CLTDTA library. Press Enter.

You may find that file F01132 exists in the CLTCOM library as well as the CLTDTA library. If so, you need to remove it from the CLTCOM library using the following command on an AS/400 command line:

```
DLTF FILE (CLTCOM/F01132)
```

Press Enter.

- e. Move the Production files F01131, F01132, F4201, F4211LA, F4301, F4311LA, F4314, F4314H, F4802, and F4802H one at a time, as necessary, from the libraries where they are currently located to the CLTDTA library, using the following command on an AS/400 command line:

```
MOV OBJ (library/file) OBJTYPE(*FILE) TOLIB (CLTDTA)
```

Here *file* is the production file to be moved, and *library* is the name of the library where the file is currently located. Press Enter.

6. Following the installation of PTF A73PC000X2, you must run the Post-Install jobs for A73PC000X2 and for previous coexistence updates. Complete the following steps:

- a. Sign on the AS/400 system by typing user ID JDEPROD and its associated password. Press Enter.

You then see the main WorldSoftware menu as shown in Figure 102 on page 264.

```

G                                IBM Canada Ltd.
                                Master Directory

... GENERAL BUSINESS SYSTEMS      ... INDUSTRY SPECIFIC SYSTEMS
2. Address Book                   14. Distribution / Logistics
3. Electronic Mail                15. Manufacturing Systems
4. Accounts Receivable           16. Construction Systems
5. Accounts Payable              17. Energy & Chemical Systems
6. General Accounting             18. Property Management
7. Financial Reporting           19. Work Order / Service Billing
8. Modeling, Planning, Budgeting 20. Contract Management
9. Fixed Assets
10. Equipment/Plant Management    ... REPORT WRITERS
11. Human Resources Management    23. FASTR Reporting
                                   24. World Writer Reporting

Selection or command   Days to Software Expiration   00
====>

```

Figure 102. WorldSoftware menu

- b. Type G97PTF on the AS/400 command line, and press Enter. You see the PTF Post Upgrade menu shown in Figure 103.

```

G97PTF                            IBM Canada Ltd.
                                PTF Post Upgrade Menu

... UPDATE APPLICATIONS           ... CREATE FILES/RUN JOBS
2. Print Menu/DD Report           14. Rebuild DD/VO/DW
3. Control File Updates           15. Rebuild FRF and JDE Msg File
4. Create ASI Report              16. Build New Files List
5. Create Special Job Report      17. Create New Files
6. Print ASIs & Special Job Rpt   18. Rebuild Menu Word Search
7. Work With Instructions (ASIs)  19. Inquire on File Conv. Status
8. DREAM Writer Review Workbench

                                   ... REVIEW REPORTS/STARTUP
                                   23. Utilities
                                   24. Start JDE Subsystems

Selection or command   Days to Software Expiration   00
====>

```

Figure 103. PTF Post Upgrade menu

- c. Complete the menu items on menu G97PTF, according to the instructions in the *A7.3 PTF Install Workbook*.
- d. Following the completion of the G97PTF menu items, type G97UX2 on the AS/400 command line, and press Enter. You see the A73PC000X2 Special Jobs menu, shown in Figure 104.

```

G97UX2                                IBM Canada Ltd.
                                        A73-A73PC000X2 Special Jobs

... A73PC000X1
  2. Previous Coexistence Jobs
  3. System 52 File Conversion

... REQUIRED FOR ONE WORLD
 17. Coexistence Setup
 18. Generic Text Synchronization
 19. SDB Synchronization Setup
 20. Config String Synchronization

Selection or command   Days to Software Expiration   00
====>

```

Figure 104. Special Jobs (Part 1 of 2)

- e. Type option 2, and press Enter. You see the A73PC000X1 Special Jobs (continued) menu as shown in Figure 105.

```

G97UX1A                                IBM Canada Ltd.
                                        A73-A73PC000X1 Special Jobs (continued)

... A73PC000W1
  2. Failure to Generate F4016LD
  3. Failure to Generate F4104LM
  4. Failure to Generate F4105LG
  5. Failure to Generate F41061LE
... A73PC000W2
  7. No Special Jobs Are Required
... A73PC000W3
  9. F3460 Duplicate Cleanup
 10. F3002 Duplicate Cleanup
 11. Lot Number Century Conversion

... A73PC000W4
 14. CARDEX - Unique Key Post Inst
... A73PC000W5
 16. No Special Jobs Are Required
... A73PC000X0
 18. F3462 Duplicate Cleanup
 19. System 16 File Maintenance
 20. Move System 49 Tables
 21. System 37 File Maintenance
 22. System 48S File Conversion
 23. File Conversion for F3294
 24. File Conversion for F32941

Selection or command   Days to Software Expiration   00
====>

```

Figure 105. Special Jobs (Part 2 of 2)

- f. Select each of the menu items, in order, on the G97UX1A menu to complete the Post-Install jobs for previous coexistence updates.
 - g. When you finish each of the menu items on the G97UX1A menu, press F12 to return to the G97UX2 menu.
 - h. Select each of the rest of the menu items on the G97UX2 menu to complete the Post-Install jobs for coexistence PTF A73PC000X2.
7. Record the name of your existing WorldSoftware AS/400 Common library. For AS/400 coexistent environments, you can use user defined code (UDC) tables (F0004, F0004D, F0005, and F0005D) from your existing WorldSoftware libraries. Typically, these files are located in the AS/400 Common library.

When you define your installation plan, you configure the AS/400 Common data sources (for AS/400 Common - Production, AS/400 Common - Test, AS/400 Common - CRP, AS/400 Common) to reflect your AS/400 Common library name. When your plan is created, the appropriate OCM mappings are changed to point to your WorldSoftware files. The menu tables in the Control Tables data source have the same file names as in WorldSoftware, but different formats, so cannot be located in the AS/400 Common data source. In coexistent environments, the AS/400 Common data source is also used rather than the Control Tables data source to access the existing WorldSoftware next numbers tables (F0002 and F00021).

10.3.1 Installing OneWorld in a coexistence environment

To install OneWorld in a coexistence environment where WorldSoftware is already installed, complete the following steps during the normal installation. For more information on a complete OneWorld installation, see Chapter 9, “OneWorld installation and maintenance” on page 151.

1. When entering information into the Installation Planner (refer to “Entering information into Installation Planner” on page 190), click **Coexistent with World Software**. This ensures that the AS/400 Common data sources are created when you enter the Enterprise Server Information.
2. When specifying environment data load parameters (refer to step 7 on page 204) on the Advanced Parameters form, select **Coexistent Environment**, and select **Coexistent Merges**. However, if you are specifying the environment data load parameters for the Pristine environment, always deselect **Coexistent Environment** and deselect **Coexistent Merges**, because there is no corresponding Pristine environment in WorldSoftware with which to share data.
3. When prompted to setup the AS/400 Common (for example, AS/400 Common - Production) data sources (refer to step i on page 206) on the Data Source Setup form, ensure that you type the name of your WorldSoftware AS/400 Common library for Data Library Name.
4. When prompted to setup Client Access Express ODBC (refer to step v on page 206) on the Client Access Express ODBC Setup (32-bit) form for your AS/400 Common (for example, AS/400 Common - Production) data sources, type the name of your WorldSoftware AS/400 Common library for Default Libraries (refer to Table 15 on page 192) on the **Server** tab. The information should appear similar to the example in Figure 67 on page 198. Type the same library name for Default package library (refer to Table 15 on page 192) on the **Package(s)** tab. The information should appear similar to the example in Figure 68 on page 199.
5. When prompted to setup the Business Data data sources (refer to step i on page 206) on the Data Source Setup form, enter the name of your corresponding WorldSoftware data library, if applicable, for Data Library Name.
6. When prompted to setup Client Access Express ODBC (refer to step v on page 206) on the Client Access Express ODBC Setup (32-bit) form for your Business Data data sources, type the name of your corresponding WorldSoftware data library, if applicable, for Default Libraries (refer to Table 15 on page 192) on the **Server** tab. The information should appear similar to the example in Figure 68 on page 199. Type the same library name for Default

package library (refer to Table 15 on page 192) on the **Package(s)** tab. The information should appear similar to the example in Figure 68 on page 199.

7. Following the running of the Release Master, you must complete the definition of the AS/400 Common OneWorld Data Sources. Complete the steps listed in “Defining OneWorld AS/400 Common Data Sources (co-existence only)” on page 211. Be sure to use your WorldSoftware AS/400 Common library if it is different from the J.D. Edwards standard library name CLTCOM.

10.3.2 Verifying the installation

Following the installation of OneWorld, you should now have a working coexistence environment. To verify your coexistence environment, complete the following steps:

1. Verify which libraries contain the Generic Text files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164LA, and F00165. For each file, enter the following command on an AS/400 command line:

```
WRKOBJ OBJ(*ALL/file) OBJTYPE(*FILE)
```

Here, *file* is the Generic Text file. Press Enter.

If any of these files is in both the CLTCOM and CLTDTA libraries, delete it from the CLTCOM library by entering the following command on an AS/400 command line:

```
DLTF FILE(CLTCOM/file)
```

Here, *file* is the Generic Text file to be removed from library CLTCOM. Press Enter.

2. Verify that file F01132 exists in the CLTDTA library only by entering the following command:

```
WRKOBJ OBJ(*ALL/F01132) OBJTYPE(*FILE)
```

Press Enter.

If file F01132 is in the CLTCOM library, delete it from the CLTCOM library by entering the following command:

```
DLTF FILE(CLTCOM/F01132)
```

Press Enter.

3. In order for coexistence to work, the Generic Text files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164A, and F00165 and the Production files F01131, F01132, F4201, F4211LA, F4301, F4311LA, F4314, F4314H, F4802, and F4802H must be in the CLTDTA library in the WorldSoftware format, with triggers attached. The OneWorld installation may replace or create these files in the CLTDTA library with a OneWorld format. Complete the following steps:

- a. Verify the correct format for each of the Generic Text files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164A, and F00165 and the Production files F01131, F01132, F4201, F4211LA, F4301, F4311LA, F4314, F4314H, F4802, and F4802H with the following command:

```
DSPFD FILE(CLTDTA/file)
```

Here, *file* is name of the Generic Text or Production file to be verified. Press Enter.

If the file is formatted to OneWorld, the Data Base File Attributes section showing on the display will contain an SQL File Type of TABLE.

If there is no SQL File Type in the Data Base File Attributes section, scroll down to the bottom of the file description. Verify that a format exists for this file and that the format starts with an “I”. For example, the file description of F9801 would have a format of I9801 in the WorldSoftware format.

If the file has an “I” format and does not have an SQL File Type, it is formatted to WorldSoftware.

- b. For each of the Generic Text files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164A, and F00165 and Production files F01131, F01132, F4201, F4211LA, F4301, F4311LA, F4314, F4314H, F4802, and F4802H in the CLTDTA library that has a OneWorld format instead of the required WorldSoftware format, perform the following steps:

- i. Rename the OneWorld file using the following command and pressing Enter:

```
RNMOBJ OBJ(CLTDTA/file) OBJTYPE(*FILE) NEWOBJ(OWfile)
```

Here, *file* is the name of the Generic Text or Production file, which is in the OneWorld format.

- ii. Copy the WorldSoftware file format from the JDFDATA library to the CLTDTA library using the command and pressing Enter:

```
CRTDUPOBJ OBJ(file) FROMLIB(JDFDATA) OBJTYPE(*FILE) TOLIB(CLTDTA)  
DATA(*YES)
```

Here, *file* is the name of the Generic Text or Production file for which the WorldSoftware format is required.

- c. Verify that each of the Generic Text files F0016, F00161, F00162, F00163, F00163LA, F00164, F00164A, and F00165 has a trigger using the following steps:

- i. Type:

```
DSPFD FILE(CLTDTA/file)
```

Here, *file* is name of the Generic Text file to be verified. Press Enter.

- ii. Scroll down four times to where trigger information should be displayed.
- iii. If no trigger information exists in the file description, the triggers have not been set. If no triggers exist, you must return to the G97UX2 menu (refer to step d on page 264) and run each of the options again.

- d. Verify that each of the files F3294, F32941, and F4314H has a trigger using the following steps:

- i. Type the command:

```
DSPFD FILE(CLTDTA/file)
```

Here, *file* is name of the file to be verified. Press Enter.

- ii. Scroll down four times to where trigger information should be displayed.
- iii. If no trigger information exists in the file description, the triggers have not been set. If no triggers exist, you must return to the G97UX2 menu (refer to step d on page 264) and run each of the options again.

10.4 Converting OneWorld environment to coexistence

In certain situations, some AS/400 OneWorld customers want to install specific WorldSoftware applications to coexist with their OneWorld applications. This section reviews the process for doing so, as well as some of the pitfalls that can result from this process.

Customers may find that converting the existing OneWorld AS/400 environment to a coexistence environment will result in increased functionality and performance of certain modules or provide some other benefits.

The procedures in this section describes how to create a OneWorld/WorldSoftware coexistence configuration by performing a fresh installation of WorldSoftware on an existing OneWorld AS/400 installation.

Note

If you have both a OneWorld and a WorldSoftware installation *active*, it is not possible to combine the two environments into a coexistence environment.

Conversion procedure scenarios

When you convert to a coexistence environment, you may not need to preserve the data that was input into the OneWorld environment. For example, the OneWorld data that was input may be from testing and does not need to be saved.

This section reviews a procedure for converting to a coexistence environment *and* saving the existing OneWorld data. If you want to convert to a coexistence environment and do *not* need to save your existing OneWorld environment, you should remove your OneWorld environment, install WorldSoftware, and then install OneWorld as discussed in 10.3, “Converting WorldSoftware environment to coexistence” on page 262.

Table structure differences

Table structure is the only coexistence issue discussed in this section. OneWorld and WorldSoftware do not have the same table definitions. In fact, some of the table attributes are distinctly different. This is because WorldSoftware is written using the RPG programming language, while OneWorld is written using the OneWorld case tool, which is based on the C programming language.

Additionally, WorldSoftware can read data from tables created with RPG but cannot read data from tables created with the OneWorld case tool. OneWorld, however, can read data from tables created with RPG as well as the OneWorld case tool. Because of these differences, it is necessary to carefully plan for a coexistence environment.

Converting the entire configuration

This section provides information for converting the entire configuration to coexistence. Once you are beyond this technical hurdle, all coexistence documentation, procedures, and policies can carry you forward without special circumstances or considerations. There are many other options available when converting the existing database to coexistence, although these are not discussed in this document. For example, functionality or performance gains may

be realized if you convert only subsets of some tables. However, with this method, there are many support and upgrade ramifications.

10.4.1 Reading and converting data

A typical coexistence installation is carried out when WorldSoftware is already installed. In such case, OneWorld is installed on top of the existing data libraries. If certain tables exist, the original WorldSoftware table structures remain intact. If there are new OneWorld-only tables, they are created in the existing data libraries but can only be accessed by OneWorld applications.

The installation procedure discussed in this section deals with the opposite situation, where OneWorld is installed before WorldSoftware is installed. In this case, the OneWorld installation must be removed, then WorldSoftware installed, and then the OneWorld pieces are placed on top of the WorldSoftware installation. Because OneWorld can read data from OneWorld and WorldSoftware tables, while WorldSoftware can read data only from WorldSoftware tables, coexistence is a one-way street.

When installing WorldSoftware after OneWorld, the WorldSoftware tables do not already exist so you must convert the table definitions from their existing OneWorld format to the WorldSoftware format. Since programs do not exist to perform this conversion, existing tools must be used. At a high-level, the new WorldSoftware table definitions are installed. These tables are populated with existing OneWorld data, and then certain installation steps are performed again.

Libraries

The default libraries are different in a WorldSoftware and OneWorld environment, and are intermixed in coexistence environment. The three main libraries this section discusses are the Control Tables library (TESTCTL), the Common library (TESTCOM), and Business Data library (TESTDTA). Table 20 addresses these three libraries and the three configurations.

Table 20. OneWorld, WorldSoftware, and coexistence libraries

Library	OneWorld	WorldSoftware	Coexistence
TESTCTL	- Next Numbers - User Defined Codes - OneWorld Menus ¹		- OneWorld Menus ¹
TESTCOM		- Next Numbers - User Defined Codes - WorldSoftware Menus ¹	- Next Numbers - User Defined Codes - WorldSoftware Menus ¹
TESTDTA	- All other OneWorld Business Data tables	- All other WorldSoftware Business Data tables	- All other WorldSoftware Business Data tables - All other OneWorld Business Data tables not already defined in WorldSoftware

Library	OneWorld	WorldSoftware	Coexistence
1. A distinction is made between the OneWorld and WorldSoftware menu, because they aren't coexistent tables, but rather cohabitation tables. The tables share the same naming conventions and the same basic table formats, but the data inside the tables is not coexistent.			

The WorldSoftware tables are created with the correct format by the WorldSoftware installation process. The business data tables can be populated with existing OneWorld data. However, the User Defined Code and Next Numbers tables must be merged with the existing WorldSoftware data.

10.4.2 Converting to a coexistence environment and saving the data

When you convert to a coexistence environment, you may need to preserve the data that was input into the OneWorld environment. This section provides a procedure for converting to the coexistence environment and preserving your OneWorld environment data.

Consider these points regarding the procedure:

- It needs to be performed for each environment you convert.
- It uses the library names indicated in Table 20. You must substitute your library names for these.
- It uses the J.D. Edwards standard data source naming conventions.
- During most of this conversion, the OneWorld environment is inaccessible.
- All steps that involve AS/400 commands should be executed from a session on which QSECOFR has signed on.
- All steps that involve OneWorld should be performed from the Deployment Server as user JDE in the Planner Environment (JDEPLAN).

The procedure is explained in the following steps:

1. Perform a complete backup of your Deployment Server (directory structure and database), the AS/400 libraries, and the integrated file system on the AS/400 system.
 - a. Sign on the AS/400 system as QSECOFR.
 - b. Type `GO SAVE` on an AS/400 command line, and press Enter to access the AS/400 SAVE menu.
 - c. Enter option 21, and press Enter.
2. Check the Minimum Technical Requirements for the most current OneWorld and WorldSoftware coexistence requirements and ensure that you can meet them.
3. Perform the WorldSoftware installation (refer to the *A7.3 Install Workbook*). Use the standard WorldSoftware data library name (CLTDTA) for the DTA lib in the WorldSoftware plan.
4. Install the WorldSoftware Cumulative PTF (refer to the *A7.3 PTF Install Workbook*).
5. Install the WorldSoftware Coexistent PTF (refer to the *A7.3 PTF Install Workbook* and the Coexistence PTF notice).

6. Rename the data library (TESTDTA) to a library with a different name (TESTDTAOW). On an AS/400 command line, type:

```
RNMOBJ OBJ(TESTDTA) OBJTYPE(*LIB) NEWOBJ(TESTDTAOW)
```

Press Enter.

7. The Next Numbers must be merged from the existing OneWorld Control Tables library (TESTCTL) to the new WorldSoftware common Library (TESTCOM). This merges any new OneWorld Next Numbers into the WorldSoftware tables and updates any duplicate key records with information from the existing OneWorld Next Numbers tables. Execute the following commands:

- a. Sign on to the AS/400 system as QSECOFR.

- b. On an AS/400 command line, type:

```
CPYF FROMFILE(TESTCTL/F0002) TOFILE(TESTCOM/F0002) MBROPT(*UPDADD)  
FMTOPT(*NOCHK)
```

Press Enter.

- c. On an AS/400 command line, type:

```
CPYF FROMFILE(TESTCTL/F00021) TOFILE(TESTCOM/F00021) MBROPT(*UPDADD)  
FMTOPT(*NOCHK)
```

Press Enter.

8. The User Defined Codes must be merged from the existing OneWorld Control Tables library (TESTCTL) to the new WorldSoftware common Library (TESTCOM). This merges any new OneWorld User Defined Codes into the WorldSoftware tables and updates any duplicate key records with information from the existing OneWorld User Defined Code tables (in essence, saving your previously entered User Defined Codes). Type the following commands:

- a. CPYF FROMFILE(TESTCTL/F0004) TOFILE(TESTCOM/F0004) MBROPT(*UPDADD)
FMTOPT(*NOCHK)

Press Enter.

- b. CPYF FROMFILE(TESTCTL/F0004D) TOFILE(TESTCOM/F0004D) MBROPT(*ADD)
FMTOPT(*NOCHK)

Press Enter.

- c. CPYF FROMFILE(TESTCTL/F0005) TOFILE(TESTCOM/F0005) MBROPT(*UPDADD)
FMTOPT(*NOCHK)

Press Enter.

- d. CPYF FROMFILE(TESTCTL/F0005D) TOFILE(TESTCOM/F0005D) MBROPT(*ADD)
FMTOPT(*NOCHK)

Press Enter.

9. Perform the following steps for *each* of your OneWorld machine names: *machine*-B733 Server Map (where *machine* is the name of your AS/400 system), System-B733, and Local:

- a. Create a new OneWorld data source that points to the library renamed in step 6. For example, Business Data - TESTOW should be created that points to the TESTDTAOW library.

- i. Type GH9611 in Fast Path, and press Enter to display the Advanced Operations menu.

- ii. Double-click **Database Data Sources P986115**.
- iii. Double-click the machine name for which you are creating this data source.
- iv. Click **Add**.

You see a form similar to the display in Figure 62 on page 196.

- v. Type `Business Data - TESTDTAOW` for Data Source Name.
- vi. Type `TESTDTA` for Library Name.
- vii. Type your AS/400 name for Library List Name.
- viii. Type `Business Data - TESTDTAOW` for Database Name.
- ix. Type your AS/400 name for Server Name.

Otherwise, type and make selections for the other fields so your form looks like the display in Figure 62 on page 196.

- x. Click **OK**.
- xi. Perform steps 4 on page 202 through 18 on page 202 to setup the ODBC Data Source, except that the Data Source Name is Business Data - TESTDTAOW, and the Default Libraries and Default Package Library are TESTDTAOW.

- b. Add a DEFAULT record in the Database Parameters section of OneWorld data sources.

- i. Click **Database Params** on the Form Exit menu.
- ii. Type `Business Data - TESTDTAOW` for Data Source.
- iii. Type `B7332` for Release.
- iv. Type `DEFAULT` for Object Name.
- v. Click **Next**.

- c. Verify that the new data source DEFAULT record exists and has Object Name set to uppercase `DEFAULT`, Copy Data set to `Y`, and Create Tables set to `1`.

- i. Type `Y` for Copy Data.
- ii. Type `1` for Create Tables.
- iii. Click **OK**.

10. Verify that the AS/400 Common - *environment* data sources point to the correct Local data source (OneWorld Planner - B73x) library, where *environment* is the OneWorld environment for which you are creating this coexistence environment. For example, AS/400 Common - Test should point to the TESTCOM library. Ensure that the Data Source Name and Database Name are identical. For example, ensure that there are the same number of spaces between AS/400 and Common in both the Data Source Name and Database Name.

11. If it exists, verify that the AS/400 Common - *environment* data sources point to the correct System data source (System - B73x) library, where *environment* is the OneWorld environment for which you are creating this coexistence environment. For example, AS/400 Common - Test should point to the TESTCOM library. If it does not exist, create this data source by copying the Business Data - *environment* data source, making the appropriate changes.

12. If it exists, verify that the AS/400 Common - *environment* data sources point to the correct Server Map data source (Machine Name - Server Map) library, where *environment* is the OneWorld environment for which you are creating this coexistence environment. For example, AS/400 Common - Test should point to the TESTCOM library. If it does not exist, create this data source by copying the Business Data - *environment* data source and making the appropriate changes.
13. Verify that the OCM mappings for the Next Numbers tables (F0002 and F00021) and the User Defined Code tables (F0004, F0004D, F0005, and F0005D) for all Local data source (OneWorld Planner - B733) environments point to AS/400 Common - *environment* data sources.
 - a. Double-click **Object Configuration Manager** P986110 from the Advanced Operations menu.
 - b. Double-click **Local**.
 - c. Type the name of your OneWorld environment for which you are creating this coexistence environment in Environment, and click **Find**.
 - d. For each table not listed, complete the following steps:
 - i. Click **Add**.
 - ii. Complete the form so it looks like the example in Figure 106, except for the table name and the environment name.

Environment Name	<input type="text" value="PRT733"/>
Object Name	<input type="text" value="F0002"/>
Primary Data Source	<input type="text" value="AS/400 Common"/>
User	<input type="text" value="PUBLIC"/>
Object Type	<input type="text" value="TBLE"/> Table Definition
Data Source Mode	<input type="text" value="P"/> Primary
Secondary Data Source	<input type="text"/>
Allow QBE	<input checked="" type="checkbox"/> All QBE Allowed

Figure 106. OneWorld OCM map creation

- iii. Click **OK**.
- iv. Click **Active/Inactive** on the Row Exit menu to make it active.
- e. For each of these tables listed, complete the following steps:
 - i. Double-click the row.
 - ii. Verify that the information appears similar to what is shown in Figure 106, and that the Primary Data Source is AS/400 Common. Make the appropriate changes.
 - iii. Click **OK**.
 - iv. If the Primary Data Source was incorrect, click **Active/Inactive** on the Row Exit menu to make it inactive.
 - v. Click **Copy** to copy this row to correct the Primary data Source.

- vi. Complete the form so it appears like the example in Figure 106, except for the table name and the environment name.
 - vii. Click **OK**.
 - viii. Click **Active/Inactive** on the Row Exit menu to make it active.
14. Verify that the OCM mappings for the Next Numbers tables (F0002 and F00021) and the User Defined Code tables (F0004, F0004D, F0005, and F0005D) for all System data source (System - B733) environments point to OneWorld Local - *pathcode* or AS/400 Common - *environment* data sources.
- a. Double-click **Object Configuration Manager** P986110 from the Advanced Operations menu.
 - b. Double-click **System - B733**.
 - c. Type the name of your OneWorld environment for which you are creating this coexistence environment in Environment, and click **Find**.
 - d. For each table not listed, complete the following steps:
 - i. Click **Add**.
 - ii. Complete the form so it appears like the example in Figure 106, except for the table name and the environment name.
 - iii. Click **OK**.
 - iv. Click **Active/Inactive** on the Row Exit menu to make it active.
 - e. For each table listed, complete the following steps:
 - i. Double-click the row.
 - ii. Verify that the information appears similar to the display in Figure 106 and that the Primary Data Source is AS/400 Common. Make the appropriate changes.
 - iii. Click **OK**.
 - iv. If the Primary Data Source was incorrect, click **Active/Inactive** on the Row Exit menu to make it inactive.
 - v. Click **Copy** to copy this row to correct the Primary data Source.
 - vi. Complete the form so it appears like the display shown in Figure 106, except for the table name and the environment name.
 - vii. Click **OK**.
 - viii. Click **Active/Inactive** on the Row Exit menu to make it active.
15. Verify that the OCM mappings for the Next Numbers tables (F0002 and F00021) and the User Defined Code tables (F0004, F0004D, F0005, and F0005D) for all Server Map data source (*machine* - Server Map) environments point to AS/400 Common - *environment* data sources.
- a. Double-click **Object Configuration Manager** P986110 from the Advanced Operations menu.
 - b. Double-click ***machine* - Server Map**.
 - c. Type the name of your OneWorld environment for which you are creating this coexistence environment in Environment, and click **Find**.
 - d. For each table not listed, complete the following tasks:

- i. Click **Add**.
 - ii. Complete the form so it appears like the display in Figure 106, except for the table name and the environment name.
 - iii. Click **OK**.
 - iv. Click **Active/Inactive** on the Row Exit menu to make it active.
- e. For each table listed, complete the following tasks:
 - i. Double-click the row.
 - ii. Verify that the information appears similar to what is shown in Figure 106 on page 274 and that the Primary Data Source is AS/400 Common. Make the appropriate changes.
 - iii. Click **OK**.
 - iv. If the Primary Data Source was incorrect, click **Active/Inactive** on the Row Exit menu to make it inactive.
 - v. Click **Copy** to copy this row to correct the Primary data Source.
 - vi. Complete the form so it appears like the example in Figure 106, except for the table name and the environment name.
 - vii. Click **OK**.
 - viii. Click **Active/Inactive** on the Row Exit menu to make it active.
16. OneWorld will no longer access the User Defined Code tables that exist in the TESTCTL library. Rename the User Defined Code tables (F0004, F0004D, F0005, and F0005D) in TESTCTL. After the conversion process, compare the two sets of tables to verify that nothing went wrong in the copy process. After a thorough test of User Defined Codes in both the WorldSoftware and OneWorld environments, and when you feel comfortable with the testing, these tables may be deleted. On an AS/400 command line, type the following commands in order, and press Enter after each one:
 - a. `RNMOBJ OBJ(TESTCTL/F0004) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F0004OR)`
 - b. `RNMOBJ OBJ(TESTCTL/F0004D) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F0004DOR)`
 - c. `RNMOBJ OBJ(TESTCTL/F0005) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F0005OR)`
 - d. `RNMOBJ OBJ(TESTCTL/F0005D) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F0005DOR)`
17. OneWorld will no longer access the Next Numbers tables that exist in the TESTCTL library. Rename the Next Numbers tables (F0002 and F00021) in TESTCTL. After the conversion process, compare the two sets of tables to verify that nothing went wrong in the copy process. After a thorough testing of Next Numbers in both the WorldSoftware and OneWorld environments, and when you feel comfortable with the testing, you can delete these tables. On an AS/400 command line, type the following commands in order, and press Enter after each one:
 - a. `RNMOBJ OBJ(TESTCTL/F0002) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F0002OR)`
 - b. `RNMOBJ OBJ(TESTCTL/F00021) OBJTYPE(*FILE) NEWOBJ(TESTCTL/F00021OR)`
18. Copy Version XJDE0021 (Refresh CRP Data from Production), and create Version TESTOW (Copy OneWorld Data to Coexistence Library).
19. Type `GH9011` in Fast Path, and press Enter to display the System Administrative Tools menu.
20. Double-click **Batch Versions** (P98305).

21. Type R98403 in Batch Application.
22. Click **Find**.
23. Select Version **XJDE0021** (Refresh CRP Data from Production), and click **Copy**.
24. Complete the form to create version TESTOW (Copy OneWorld Data to Coexistence Library). It should look similar to the example in Figure 107.

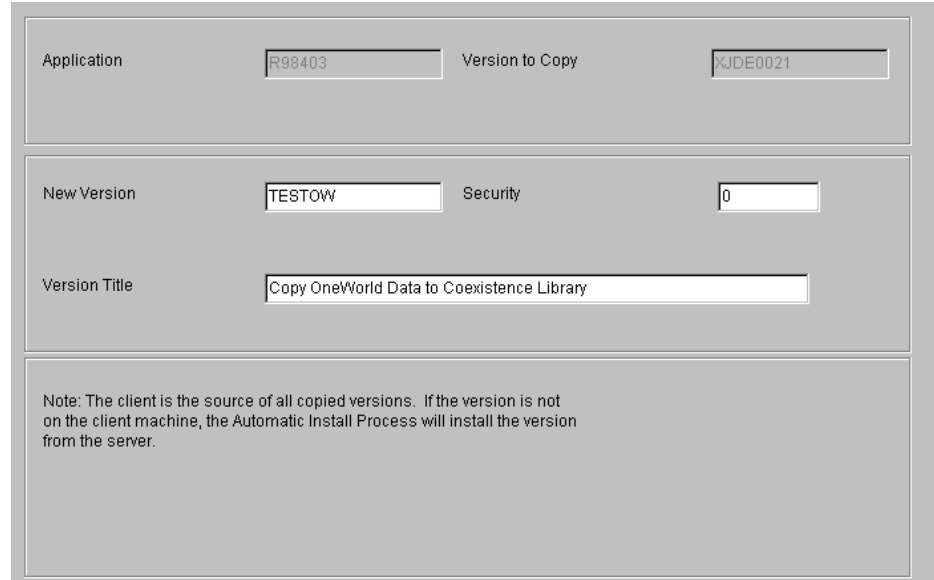
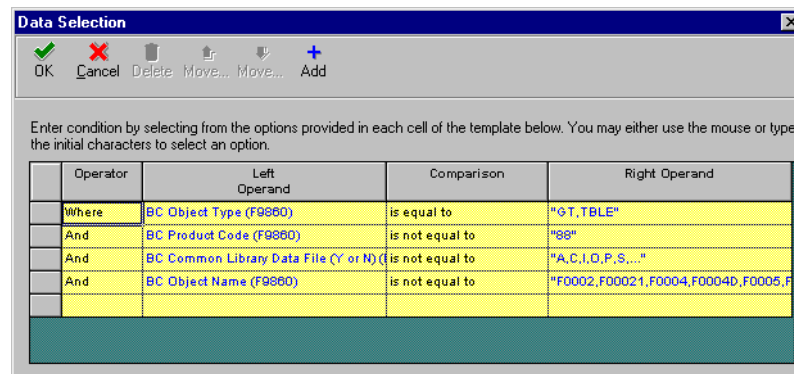


Figure 107. OneWorld batch version copy

25. Click **OK**.
26. Verify the Data Selection of your new version. Click your new version **TESTOW** to highlight it, and click **Data Selection** on the Row Exit menu. The data selection should look the example in Figure 108.

Verify that BC Object Name (F9860) is not equal to "F0002, F00021, F0004, F0004D, F0005, F0005D, F0082, F00821, F00825, F00826, F0083, F0084, F9200, F9201, F9202, F9203, F9204, F9205, F9207, F9210, F9211, GT92002, GT9860A, GT9862A".



	Operator	Left Operand	Comparison	Right Operand
	Where	BC Object Type (F9860)	is equal to	"Q T, TBLE"
	And	BC Product Code (F9860)	is not equal to	"98"
	And	BC Common Library Data File (Y or N)	is not equal to	"A.C.I.O.P.S..."
	And	BC Object Name (F9860)	is not equal to	"F0002,F00021,F0004,F0004D,F0005,F"

Figure 108. OneWorld batch version data selection

27. Make any changes, if necessary, and click **OK**.

28. The processing options need to be modified. Click your new version **TESTOW** to highlight it, and select **Processing Options** from the Row Exit menu. You see a form similar to the display in Figure 109.

The screenshot shows a software interface with four tabs: **Environment**, **Update**, **Print**, and **Licensing**. The **Environment** tab is active. Below the tabs are four numbered instructions, each followed by an input field:

- 1. Enter the Environment for the database to be created for. (If this report is called from another process, the Environment will be passed in) [Empty field]
- 2. or Enter the Data Source for the database to be created for. [Business Data - CR]
- 3. Enter a '1' to load Production Data or a '2' to load Demonstration Data. The default is to load production data. (If this report is called from another process, this flag will be passed in) [2]
- 4. Enter the source Data Source for Loading of Data. (If this report is called from another process, the source Data Source will be passed in) [Business Data - PR]

Figure 109. OneWorld batch version processing options

29. On the **Environment** tab, complete the following steps:

- Type your environment name (for example, PRD733) in field 1.
- Clear field 2.
- Type 2 in field 3.
- Type `Business Data - TESTOW` in field 4.

30. Select the **Update** tab, and complete the following steps:

- Clear field 5.
- Type 2 in field 6.
- Clear field 7.
- Clear field 8.
- Clear field 9.

31. Select the **Print** tab, and clear field 10.

32. Select the **Licensing** tab, and complete the following tasks:

- Clear field 11.
- Clear field 12

33. Click **Submit**.

34. Select **Screen**, and click **OK**.

35. You may be prompted for the JDE password (Figure 110). Type the password for JDE, and click **OK**.

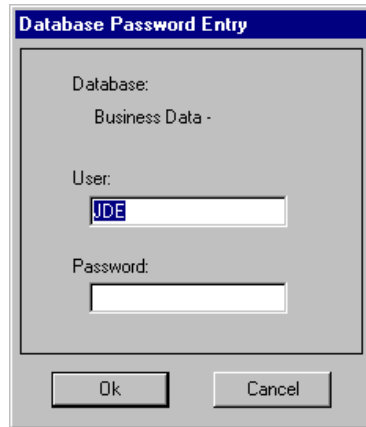


Figure 110. OneWorld Database Password Entry

36. When this UBE completes, you see a report where the only successful copies will be for new tables.
37. This UBE needs to be repeated with different processing options to copy records to existing tables.
38. The processing options need to be modified. Click your new version **TESTOW** to highlight it, and select **Processing Options** from the Row Exit menu. You then see a form similar to the display in Figure 109.
39. On the Environment tab, complete the following tasks:
 - a. Type your environment name (for example, PRD733) in field 1.
 - b. Clear field 2.
 - c. Type 2 in field 3.
 - d. Type Business Data - TESTOW in field 4.
40. Select the **Update** tab, and complete the following tasks:
 - a. Clear field 5.
 - b. Type 2 in field 6.
 - c. Clear field 7.
 - d. Clear field 8.
 - e. Type Y in field 9.
41. Select the **Print** tab, and blank out field 10.
42. Select the **Licensing** tab, and complete the following tasks:
 - a. Clear field 11.
 - b. Clear field 12
43. Click **Submit**.
44. Select **Screen**, and click **OK**.
45. You may be prompted for the JDE password, similar to the example in Figure 110. Type the password for JDE, and click **OK**.
46. When this UBE completes, you see a report where the only successful copies will be for existing tables. For tables that were created in the previous run of this UBE, the record copies fail during this run because they are duplicate records.

47. Review both reports. For those files where no records copied successfully, try using the following AS/400 command for each of these files:

```
CPYF FROMFILE(TESTDTAOW/file) TOFILE(TESTDTA/file) MBROPT(*ADD)
FMTOPT(*MAP)
```

Here, *file* is the name of the file that was previously unsuccessful. Press Enter.

If this is unsuccessful, investigate whether a copy of the file is required because the format of this file may have changed.

48. If your UDCs point to OneWorld Local - *pathcode*, you now need to recopy them from AS/400 Common - *environment* to OneWorld Local - *pathcode*. Run the appropriate version of R98403 for each path code/environment combination. For example, R98403 XJDE0013 is pre-delivered for the Production environment:

- a. Type `GH9011` in Fast Path and press Enter to display the System Administrative Tools menu.
- b. Double-click **Batch Versions** (P98305).
- c. Type `R98403` in Batch Application.
- d. Click **Find**.
- e. Select Version **XJDE0013**, and click **Copy**.
- f. Complete the form to create the TESTOW2 version (Copy OneWorld Common UDCs to Local).
- g. The processing options need to be modified. Click your new version **TESTOW** to highlight it, and select **Processing Options** from the Row Exit menu. You see a form similar to the display in Figure 109 on page 278.
- h. On the Environment tab, complete the following tasks:
 - i. Clear field 1.
 - ii. Type `OneWorld Local - TESTDTAOW` in field 2.
 - iii. Clear field 3.
 - iv. Type `AS/400 Common` in field 4.
- i. Select the **Update** tab, and complete the following tasks:
 - i. Clear field 5.
 - ii. Type `2` in field 6.
 - iii. Type `A` in field 7.
 - iv. Clear field 8.
 - v. Clear field 9.
- j. Select the **Print** tab, and clear field 10.
- k. Select the **Licensing** tab, and complete the following tasks:
 - i. Clear field 11.
 - ii. Clear field 12.
- l. Click **Submit**.
- m. Select **Screen**, and click **OK**.
- n. You may be prompted for the JDE password, similar to the display in Figure 110 on page 279. Type the password for JDE, and click **OK**.

- 49.Recompress one full package for each path code. This compresses the jdeb7.mdb for all packages within that path code. For more information, refer to Chapter 13, “OneWorld package management” on page 327.
- 50.Reinstall client machines so that they get the new jdeb7.mdb. For more information, refer to “Installing OneWorld in the interactive mode” on page 237.

Chapter 11. Client Access Express and OneWorld

Client Access Express is a TCP/IP-only, 32-bit Windows client that was added to the Client Access for Windows family in 1999. This client is tested and supported in connection to V4R2 OS/400 and later. Many of the functions will continue to work with previous releases of OS/400. However, Client Access Express relies on OS/400 V4R2 NetServer function for file and print serving.

Client Access Express is the required method for a PC with OneWorld to be connected to the AS/400 system. This is because OneWorld takes the advantage of enhanced, multi-threaded capability of the ODBC driver that is provided with Client Access Express.

This chapter offers a brief discussion of the benefits and features of Client Access Express. It also provides a brief overview of the differences between Client Access Express and Client Access/400 and the settings that are migrated during an upgrade. An overview of ODBC connectivity is provided as well as a discussion of Client Access licensing.

For in-depth coverage of Client Access topics, refer to the redbooks *AS/400 Client Access Express for Windows: Implementing V4R4M0*, SG24-5191, and *Management Central: A Smart Way to Manage AS/400 Systems*, SG24-5407. Or, visit the Client Access Web page at: <http://www.as400.ibm.com/clientaccess/>

11.1 Client Access Express functions

The following list offers a brief description of the functionality provided with Client Access Express. You can find more detailed information on specific functions in the redbook *AS/400 Client Access Express for Windows: Implementing V4R4M0*, SG24-5191.

- **Installation and migration:** Client Access Express for Windows can be installed on Windows 95, Windows 98, Windows NT 4.0, and Windows 2000. Windows 2000 requires at least Service Pack SF60698. For more information on Client Access Express on Windows 2000, see Informational APAR II11853. Client Access Express for Windows also migrates certain client configuration information from some previous Client Access products.
- **ODBC driver:** OneWorld Explorer uses the Open Database Connection (ODBC) interface to access database management systems. ODBC is a Microsoft specification that enables applications to access data using Structured Query Language (SQL) as a standard language. ODBC provides a consistent set of APIs that permit a single application to access different database management systems. The Client Access Express ODBC Driver is the only supported ODBC driver for a connection of a PC with OneWorld and the AS/400 system.
- **TCP/IP and dial-up connectivity:** Easy access to TCP/IP connectivity from each of its functions is one of the improvements in Client Access Express for Windows. It offers increased flexibility by integrating the management of the Client Access user ID and password with that of the Windows operating system. Note that only TCP/IP connectivity is supported by this client.
- **TCP/IP Secure Sockets Layer:** The Express Client uses the TCP/IP Secure Sockets Layer (SSL) Version 3.0 to encrypt the user ID and password, as well

as any other information being sent between the AS/400 system and the desktop. You can choose what is encrypted. For example, you may want to encrypt data being transmitted by Data Transfer but find it unnecessary to encrypt data being sent to your PC printer using 5250 printer emulation.

- **AS/400 NetServer:** AS/400 NetServer enables an AS/400 system to act as a file and print server in a Windows network. AS/400 NetServer has been a function of the base of OS/400 since Version 4 Release 2. Although AS/400 NetServer is not part of Client Access Express for Windows, it is used to replace the network drive and network printer functions that are part of Client Access for Windows 95 and Windows NT.
- **AS/400 Operations Navigator:** AS/400 Operations Navigator is a graphical user interface (GUI) that provides administrators and operators with an explorer-like view of the AS/400 resources. With the addition of the Management Central function, you can now manage multiple AS/400 systems using Operations Navigator. More information on Operations Navigator and Management Central are provided later in 11.2, “Operations Navigator and Management Central” on page 286.
- **Display and print emulation:** Client Access Express PC5250 is equipped with useful functions that help you work with AS/400 display and print emulation sessions. Many enhancements are added into this release to deliver improved connectivity, efficiency, usability, manageability, and flexibility to those who work with the AS/400 system using the “green-screen” environment.
- **AFP Workbench Viewer:** This product is a tool that helps you work with many kinds of print output and PC files. It helps you display a What You See Is What You Get (WYSIWYG) view of the AS/400 output and PC files, navigate through them, view many popular PC graphic formats, and create and share AFP resources for AS/400 and PC printing jobs.
- **Data Transfer:** Data Transfer allows data to be transferred between the AS/400 system and the PC based on the SQL engine. Wizards are included in the Data Transfer application, allowing users to create, run, and save transfer requests easily. In the Client Access Express, new functions, such as an Auto-run transfer request, have been added to the Data Transfer application.
- **Middleware for client/server applications:** Client Access Express delivers many middleware components that contain popular programming interfaces for you to use when developing client/server applications. You can use your development tool of choice to work with the AS/400 server effectively and efficiently.
- **Software maintenance:** Client Access Express is maintained automatically with the Check Service Level function. Several options let you set when and how those updates will be performed. It also introduces a new function which distributes any kind of PC application with the support of add-ins. You can add your applications in the Express Client installation process. An add-in can consist of files, user written programs or product setup programs. Once installed, the add-ins are maintained as an Express client component by the Check Service Level.
- **Administering Client Access Express:** Client Access Express for Windows provides you with tools for administering the Express client. For example, you can import or export AS/400 connections or backup and restore Express client configurations. Furthermore, for administrators for whom security has always

been a major issue, Policies and Application Administration assists you in managing functions used by your network of PCs. Client Access Express takes full advantage of System Policy. Each time a PC is started, system policies can be downloaded from a server to the PC and enforced through policies defined by an administrator. Many templates are provided to offer an outstanding granularity in the functions an administrator can control. The users can be limited in areas such as the number of 5250 sessions started and prevented from creating a new file. However, they can be allowed to append data to an existing file, but prevented from installing an Operations Navigator subcomponent. When an Express client user connects to an AS/400 system, Application Administration checks the AS/400 user profile to determine which functions can be used. An AS/400 administrator defines a security setting to allow or deny users or groups of users access to Operations Navigator, PC5250, Data Transfer, ODBC, OLE DB, and Remote Command functions.

- **Remote command:** An AS/400 system can run a command on a PC that has Client Access Express for Windows installed. A PC can start a remote command on an AS/400 system.
- **Windows Terminal Server:** Currently Client Access Express supports all functions running on Windows Terminal Server, except the Incoming Remote Command (there are no future plans to support this). For details on the available support for and the restrictions on running Client Access Express for Windows and of using AS/400 NetServer on a Windows Terminal Server, refer to Informational APAR II11373.
- **Operations Console:** Operations Console allows you to emulate the AS/400 console and the AS/400 control panel on a PC. Operations Console actually consists of two independent functions: the console emulation and the control panel. One can be used without the other. Console emulation is available for OS/400 V4R3 and later and can be used locally or remotely.
- **National Language Support:** Similar to other OS/400 licensed programs, Client Access Express for Windows supports many national language versions. The conversion process is totally transparent to the user and switching from one language to another language is as simple as clicking the mouse.
- **Client Access Diagnostic Tools:** Client Access Express ships with a set of diagnostic tools. If a user experiences problems with Client Access functions, these tools allow the collection of messages and trace information.
- **Support for plug ins:** Plug-in support provides the capability to plug-in custom tools and applications by adding new folders and objects, or context menu items to existing folders and objects to the Operations Navigator hierarchy. This enables programmers to create new dialogs for an object or launch applications from existing folders in the Operations Navigator. For example, when you right-click on an object, Operations Navigator calls the plug-in to obtain the context menu items for the object and displays it.

J.D. Edwards provides a plug-in so you can launch OneWorld from Operations Navigator.

The Operations Navigator plug-in feature in the previous release of Client Access for Windows 95 and Windows NT only provides support for programs written in Windows C++. Enhancements are made in the Express Client to include plug-in support for Java and Visual Basic applications.

11.2 Operations Navigator and Management Central

Client Access Express Operations Navigator provides a graphical user interface to manage many of the basic tasks on the AS/400 system. It is a powerful tool, allowing quick access to the same functionality as the green-screen command interface, but with a Windows Explorer style interface. Figure 111 shows Operations Navigator interface.

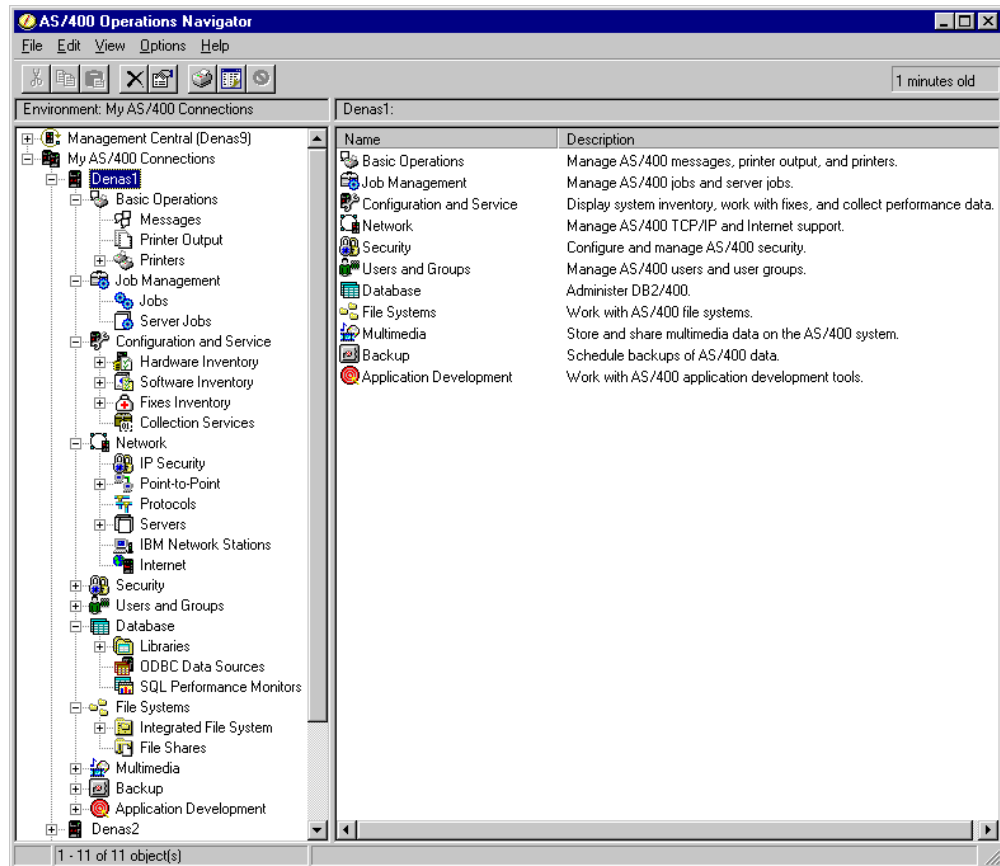


Figure 111. Operations Navigator main screen

Operations Navigator allows you to manage jobs, printers, and messages and perform many other useful tasks.

11.2.1 Operations Navigator functions

A brief description of some of the Operations Navigator functions and their uses are provided here:

- **Operations Navigator Base Support:** This sub-component provides the base support for the AS/400 Operations Navigator.
- **Basic Operations:** The Basic Operations sub-component includes functions that allow you to work with messages, printer output (spooled files), and printers on the AS/400 system. You can also allow printers to be shared with Windows clients in your network through AS/400 NetServer.
- **Job Management:** The management of user and server jobs on the AS/400 system is provided by this sub-component.

- **System Configuration:** You need to install this sub-component to view the existing hardware (including the operational status of the hardware resources) and software (including the version and release level) installed on your current AS/400 system.
- **Network:** Selecting this sub-component provides you with networking functions such as working with the AS/400 TCP/IP configuration, configuring new communications interfaces using wizards, working with the TCP/IP servers, and so on.
- **Security:** You can secure your AS/400 system by going through a security wizard provided by this option. It also allows you to configure security and auditing system values.
- **Users and Groups:** This option provides the ability to administer users and groups on the AS/400 system. You can create, delete, copy, or change the user and group properties.
- **Database:** Database functions include the ability to create, edit, and display database tables; create views of data; create and run SQL scripts; and monitor SQL performance. You can also create and manage other database objects such as journals, aliases, procedures, and user-defined functions and data types.
- **File Systems:** The File System folder in the navigation tree allows you to display and manage the AS/400 file system. You can set permissions on the file system objects. It also includes the ability to create and manage AS/400 NetServer file shares.
- **Multimedia:** If you are using the Ultimedia System Facilities (USF) application on your AS/400 system, this function provides you with the capability to store and share multimedia data such as audio and video objects.
- **Backup:** The scheduling of backup tasks on the AS/400 system can be done using the Backup function in Operations Navigator. Daily, weekly, or monthly backup policies are changed easily using the properties pages.
- **Application Development:** Using the Application Development function, you can work with inter process communication, kernel message queues, and so on.
- **Application administration:** AS/400 administrators can control the availability of functions in the Operations Navigator using the Application Administration tool. You can restrict specific functions from use by individuals or groups of users. Client Access applications, such as Data Transfer, can also be restricted using Application Administration. Management Central was first introduced with AS/400 Client Access for Windows 95 and Windows NT V3R2M0. It provides the capability to manage multiple AS/400 systems or groups of AS/400 systems using a Central System.

Refer to 12.5.4, “System management with Operations Navigator and Management Central” on page 310, for more information on how to use Management Central for system management operations.

OS/400 V4R5 brings the following enhancements to Operations Navigator:

- Graphical interface to disk management
- Database enhancements (described in 21.3.6, “Database performance analysis: Operations Navigator” on page 625)
- Integrated Netfinity Server administration

11.2.1.1 Management Central

Management Central is one of the functions of Operations Navigator that includes a suite of systems management applications that make managing multiple AS/400 systems as easy as managing a single system. Through Management Central, you can perform a wide variety of tasks from inventory collection and task scheduling to PTF maintenance and performance analysis.

Examples of the functions include system monitoring, running remote commands, managing and distributing software fixes (PTFs), file system object packaging and distribution, performance data collection, and scheduling Management Central tasks. Figure 112 shows the Management Central interface.

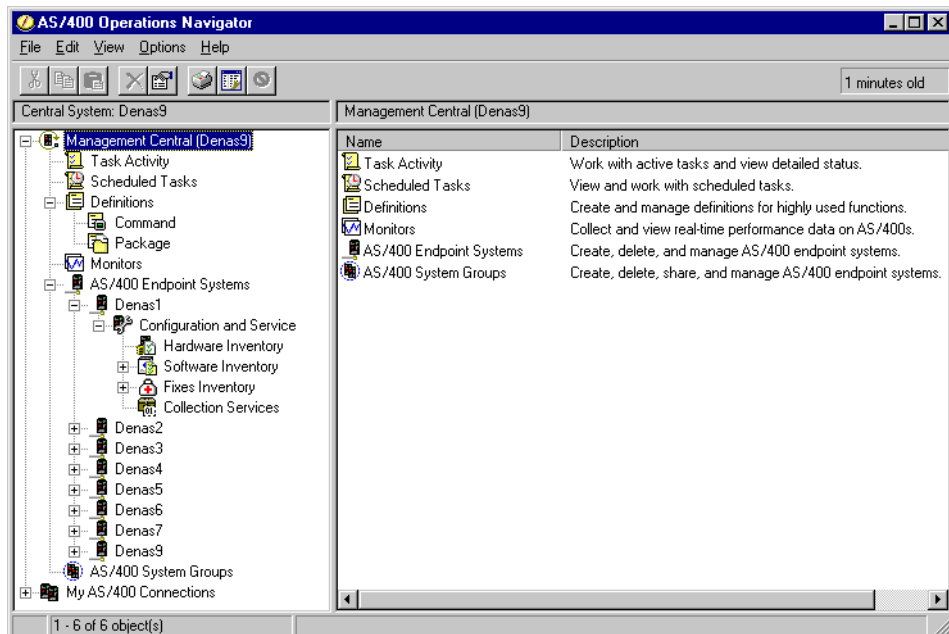


Figure 112. Management Central main screen

The functions that are covered under Management Central include:

- **Collection services:** This tool for collecting and managing performance data replaces the traditional performance monitor function with a low-overhead, automated, and on-going data collector. Data is captured with reduced system impact. Processing occurs only if and when needed. Additionally, collection services lets you control what data is collected and how that data is managed.

Each type of data supported by collection services can be controlled individually without data loss or affecting the collection of other data.

A compatible performance monitor database is created based on the data in the management collection object. However, you can defer the creation of the database until a later time.

Collecting performance sample data is enhanced by:

- Reducing the impact of collecting performance data, especially on large systems
- Allowing flexibility in the data collected
- Simplifying the management of performance data
- Promoting automated, continuous data collection

- **Object packaging:** The object packaging and distribution graphical interface provides an easy way to send objects from any file system to one or more AS/400 systems in a network. You can also restore objects, take snapshots of the objects, version packages of objects, and post execution of commands. All of these functions can be performed on a group or network of AS/400 systems and be scheduled to occur at a time most convenient for your staff.
- **PTF management for a distributed environment:** If managing PTFs among several AS/400 systems is too complicated, the new PTF management wizards are for you. The easy-to-use wizards walk you through comparing the PTF levels of multiple AS/400 systems to a model system that has a proven set of PTFs already installed. You then distribute and install any missing PTFs on the remote AS/400 systems by simply identifying the system or group of systems to be updated. You can run AS/400 commands as part of completing PTF installations or as part of normal day-to-day operations.
- **Inventory for multiple systems:** With the new graphical interface, you can schedule regular inventory collections of hardware, software, and PTF information for a group or network of AS/400 systems. From the data collected, you can search for a specific piece of information, export the information to a PC application for analysis, or just compare information in multiple systems.
- **Remote Command:** This function provides the ability to run an AS/400 CL command on any AS/400 system on the network without going through the traditional green-screen session.

11.2.2 Operations Navigator tips and techniques

Operations Navigator is a powerful tool, with many useful functions. The following list offers a few ideas and helpful hints on how you can use Operations Navigator or Management Central to assist in your routine work. There are many other features besides those mentioned here. For more details on the functions and use of Operations Navigator or Management Central, refer to the redbooks, *AS/400 Client Access Express for Windows: Implementing V4R4M0*, SG24-5191, and *Management Central: A Smart Way to Manage AS/400 Systems*, SG24-5407.

- **Drag and drop print files to the desktop:** In Operations Navigator, you can work with print files and drag them to the desktop or another destination. They are automatically converted to an ASCII format usable by your PC. Refer to 12.5.4, “System management with Operations Navigator and Management Central” on page 310, for an example of how to do this.
- **Working with print queues:** Operations Navigator can be used to quickly clean out print queues and easily delete selected print files. Refer to Figure 123 on page 312 for an example of how to do this.

You can also access another users print output by specifying another user or multiple user IDs (for example, user ID ONEWORLD) on the options pull-down menu.

- **Clean up OneWorld log and debug files:** Through Operations Navigator, files in the integrated file system can be accessed in a Windows Explorer-like view. This is an easy way to clean up OneWorld log and debug files, or clean up any other directories in the IFS. Refer to Figure 124 on page 313 for an example of how to do this.
- **Run SQL from Operations Navigator:** Even if an AS/400 system does not have the licensed program DB2 Query Manager and SQL Development Kit for AS/400, you can still run SQL queries through the Run SQL feature of Operations Navigator. Refer to Figure 125 on page 314 for an example of how to do this. This feature also has options to turn on the debug and view the job log of AS/400 host server jobs.
- **Copy shortcuts for commonly used tasks to the desktop:** Most of the tasks in Operations Navigator can be copied to the desktop for easy access. If you find yourself repeatedly using the same icon in Operations Navigator, copy it to the desktop and double-click it from there in the future. Note that this starts another instance of the Operations Navigator executable, so take care to close out the session when finished to preserve PC system resources.
- **Manage PTFs on multiple machines:** Management Central allows you to automatically install PTFs to multiple AS/400 systems to make them match a model system. This enables you to easily keep multiple systems synchronized. Refer to Figure 132 on page 319 for an example of how to do this.

11.3 Client Access Express and Client Access/400

Client Access Express is a new product and should not be confused with Client Access/400. Much of Client Access Express software has been rewritten to provide added functionality and other improvements. This section discusses some of the differences between Client Access Express and Client Access/400, as well as a discussion of what settings from a previous version of Client Access are migrated when an upgrade to Client Access Express is performed.

11.3.1 Benefits of Client Access Express

Client Access Express has many new features and enhancements that improve functionality, performance, administration, and ease of use over Client Access/400. The following list offers a brief description of some of the new and changed functions provided in this Client Access Express. For a complete listing of all the new features, go to the Client Access Express Snapshot page at:

<http://www.as400.ibm.com/clientaccess/express.htm#snapshot>

- **Improved ODBC support:** The ODBC driver in Client Access Express has been improved compared to Client Access/400 with better support for multi-threaded applications and multiprocessor PCs. Today, Client Access Express is the only supported Client Access client with OneWorld.
- **Windows 2000 support:** Client Access Express is the only version in the Client Access family that provides Windows 2000 support. For more complete information on support of Windows 2000, visit the Client Access Express Web

site at: <http://www.as400.ibm.com/clientaccess/express.htm> or refer to Informational APAR II11853

- **Installation:** The installation process has been changed to allow more flexibility and control, so system administrators can more closely control the user environment. There is more granularity in the selection process to allow a smaller footprint and control of what end users can install. For example, ODBC is now under a Data Access installation component instead of being in the base component. Therefore, an administrator can restrict users from installing ODBC.
- **Processor and memory usage:** There have been significant changes in the way Client Access Express uses memory, compared with Client Access/400. Background tasks are no longer active unless there is an active connection to an AS/400 system. Also, daemons and background tasks for file and print serving are no longer used, since this functionality is provided by the AS/400 NetServer. These changes allow Client Access Express to have a smaller memory footprint than previous versions.
- **Connectivity:** Client Access Express connects over TCP/IP only. SNA and other protocols are no longer used. This allows communications to be optimized for TCP/IP.

Connection configuration has also been simplified. You no longer need to configure connections before running your applications. This means that you can install the product and immediately run PC5250 emulation or Data Transfer just by specifying the name of the AS/400 system to which you want to connect.

- **Security:** TCP/IP network security is provided through the use of Secured Sockets Layer (SSL). SSL can be used for the entire connection or for specific functions (such as PC5250 or Data Transfer).

There are also new options for signon. You can use the Microsoft Windows logon user ID and password to sign onto the AS/400 system. If your AS/400 user ID and password is the same as your Windows desktop user ID and password, you will never be prompted for logon if you use this option.

For users who share a workstation, an option has been added to always prompt for user ID and password when opening a new application such as PC5250 or Data Transfer.

- **Data Transfer:** A wizard function has been added to the Data Transfer to AS/400 to create an AS/400 database file from an existing PC file. By specifying an existing PC file, the wizard scans the PC file and generates a definition for an AS/400 database file that can be created. The user can modify this definition and then create an AS/400 database file. Once the file is created, the user can send the PC data to the newly created AS/400 file.

A checkbox was added to the GUI to auto-run the transfer when it is opened. This enables users to run a Data Transfer without any “user” intervention by pressing the Transfer button on the GUI.

Client Access Express supports larger row sizes. Rows up to 32 KB are now supported. Previously the limit for row sizes was 4 KB.

Support for inserting into existing HTML files. This support allows users to place database data into an existing HTML file with a “template” section laid out for transfer. For example, this lets users have a Web page of text with a template section containing database data. We would only write within the

template area designated for transfer. The rest of the Web page text would remain intact.

- **Application enablement:** The Express client supports 32-bit applications only (this includes ODBC as well). For more information, refer to Web site at: <http://www.as400.ibm.com/clientaccess/toolkit/>

- **Operations Navigator:** Client Access Express offers many enhancements to Operations Navigator, including a SQL performance monitor. This allows you to monitor database performance through Operations Navigator.

You also have the capability to run SQL scripts from Operations Navigator and create SQL Indexes and stored procedures.

Management Central has new system group functions added to its hierarchy for Object packaging and Distribution, Remote Operations, PTF, Inventory, and Job Scheduler. Management Central functionality requires the use of a V4R4 OS/400 Central System. V4R3 and V4R4 OS/400 endpoint systems are supported through a V4R4 OS/400 Central System.

Client Access Express Operations Navigator now also allows users to drag and drop files to and from the AS/400 system.

The following functions, which are part of /400, are *not* part of Client Access Express:

- NS/Router (or any other SNA/APPC connectivity)
- Support to run 16-bit applications (this includes ODBC as well)
- File and print serving functions (replaced by NetServer)
- Communication Console (replaced by Operations Console)
- Graphical Access for AS/400
- Some Windows 95 and Windows NT client APIs

11.3.2 Transition from Client Access/400 to Client Access Express

Client Access Express and Client Access/400 cannot coexist on the same PC. However, Client Access Express for Windows supports migrating configuration information from Client Access for Windows 95 and Windows NT, versions V3R1M3 and V3R2M0. The following information is migrated from one client to the other:

- All Connection environments and TCP/IP configured connection within those environments are migrated and managed using Operations Navigator.
- All Directory update entries are migrated. Some modifications may be necessary to the entries following the migration because of the network drive support for Client Access Express for Windows provided by AS/400 NetServer.
- The password expiration information, password caching, and incoming remote command options are migrated.
- The Check Service Pack Level properties are migrated.
- The ANSI code page, OEM code page, and EBCDIC CCSID values are migrated.
- The properties for logging and tracing Client Access functions are migrated.
- Client Access desktop icons that have a default view of Operations Navigator, PC5250 Emulation, or User Defined are migrated.

- Operations Navigator configuration information is migrated, including Directory Services and Ultimedia System Facilities.
- PC5250 display/print emulation profiles, macros, keyboard mappings, and popup keypad configurations are migrated.
- AFP Workbench View configuration files are migrated.
- The Fax option configuration value is migrated.

11.4 ODBC

J.D. Edwards One World Explorer uses the Open Database Connection (ODBC) interface to access database management systems including DB2 UDB for AS/400. ODBC is a Microsoft specification that enables applications to access data using Structured Query Language (SQL) as a standard language. The theory behind ODBC is that any ODBC-compliant application should be able to access any database as long as someone provides a driver for that database.

ODBC provides a consistent set of APIs that permit a single application to access different database management systems. The ODBC approach is to use:

- A program separate from the application to extract database information
- A standard interface for applications to import data
- Database drivers provided by the various database vendors or third parties

The drivers are supplied as dynamic link libraries (DLLs) that an application can invoke to gain access to the database management system

Client Access Express provides drivers that can access the AS/400 database

11.4.1 ODBC interface

The ODBC interface defines a library of function calls that allow an application to:

- Connect to a database management system
- Execute SQL statements
- Retrieve results

The ODBC interface also provides:

- SQL syntax checking
- A standard set of error codes
- A standard way to connect and log on to a DBMS
- A standard representation of data types

11.4.2 ODBC components

Figure 113 on page 294 shows the ODBC components.

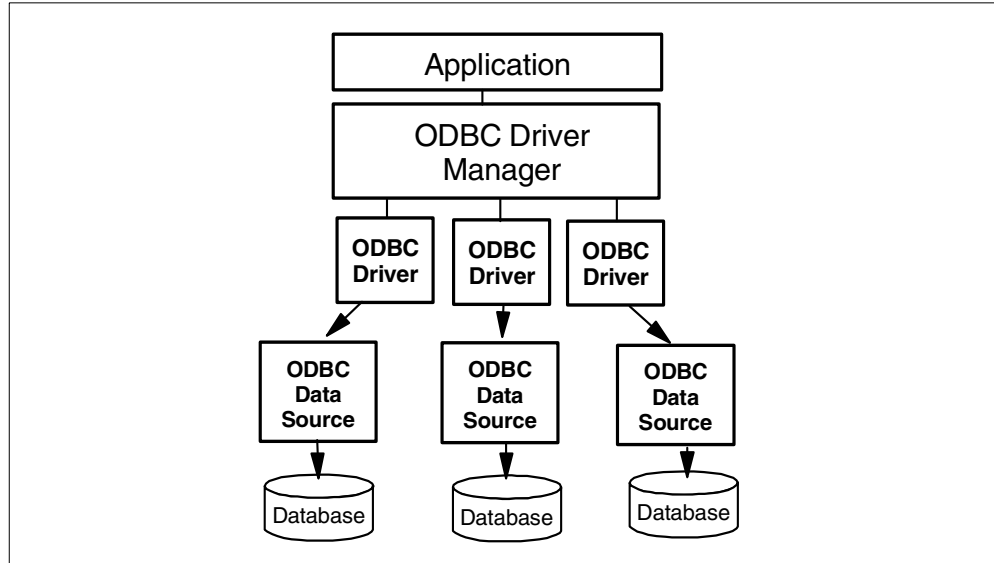


Figure 113. ODBC components

The components of an ODBC application are:

- **Application:** An application, such as OneWorld, must specifically provide ODBC capability. The application issues ODBC API calls to:
 - Allocate memory for handles
 - Request a connection
 - Create and send SQL statements
 - Request the results of the SQL
 - Report the results back to the user
 - Process any errors
 - Request commit or rollback operations for transaction control
 - End the connection
- **ODBC Driver Manager:** ODBC Driver Manager uses configured data source name to map to a specific ODBC driver. It also performs following functions:
 - Processes several ODBC initialization calls
 - Provides entry points to ODBC functions for each ODBC driver
 - Verifies that the sequence of ODBC calls being sent from the application is correct and does parameter checking
 - Installed as part of the operating system with Windows 95 and Windows NT
- **ODBC Driver:** ODBC Driver is provided by database providers and OEM vendors. For example, IBM provides it with Client Access Express. It offers these functions:
 - Receives the request to establish a connection to the data source and starts the connection
 - Submits requests to the data source
 - Translates data to other formats if requested by the application
 - Returns results to the application
 - Formats errors and returns them to the application

- Declares and manipulates cursors if necessary
- Initiates transactions if the data source requires explicit transaction initiation
- **Data Source:** Data source must be configured by the user or sometimes by applications. It specifies this information:
 - Name for the data source
 - ODBC driver to use
 - Server connection to use to access the database
 - Other database specific options

11.4.3 Job flow of an ODBC connection with Client Access Express

To illustrate how the different components of ODBC interact, this section takes you through a high-level overview of how an ODBC connection is made to the AS/400 system. An overview diagram is shown in Figure 114.

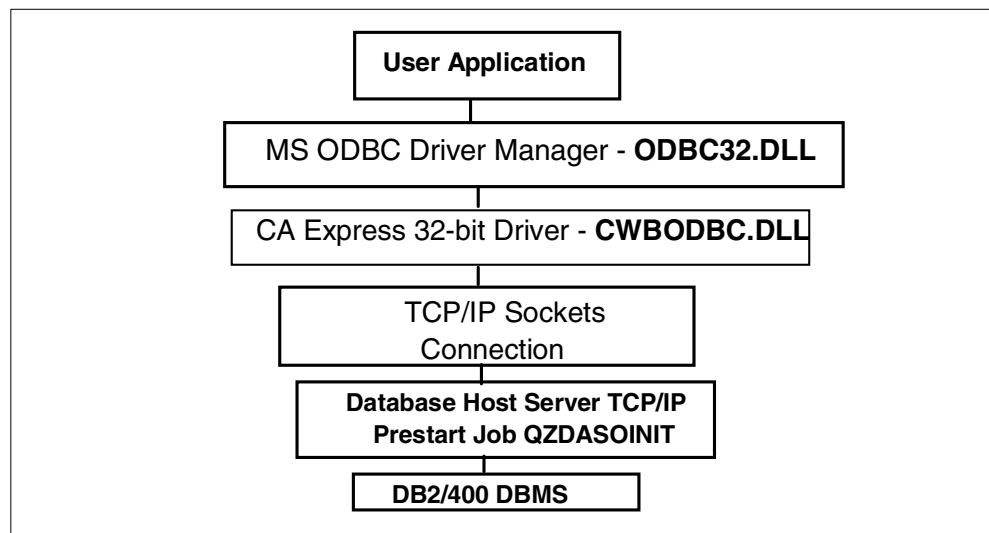


Figure 114. ODBC connection job flow

Each of the components in Figure 114 are explained here:

- **Application (for example OneWorld Explorer)**
 - Issues ODBC API calls that allocate memory for environment and connection handles (SQLAllocEnv and SQLAllocConnect)
 - Issues a ODBC API call to connect to an ODBC Data Source (either SQLConnect, SQLDriverConnect or SQLBrowseConnect)
- **ODBC Driver Manager**
 - Uses the ODBC.INI registry information to locate the correct ODBC driver, in this case the AS/400 Client Access Express ODBC driver (CWBODBC.DLL)
 - Loads the ODBC driver and data source specific connection information
- **ODBC driver starts the Client Access Express connection**
 - Verifies security and connects to a database host server job:

- Attaches to a signon server prestart job and verifies AS/400 user ID and password.
- Attaches to a database host server prestart job in the QSERVER subsystem (QZDASOINIT).

– Database host server prestart job

Runs under QUSER user profile when the job is in the prestart job pool.

When the connection is made, the QZDASOINIT job swaps to the user profile signed on to the Client Access Express ODBC connection. The job also updates its library list, sets the naming conventions, and uses an SQL package as set up in the ODBC Data Source on the client.

At this point, the ODBC connection between the PC and the AS/400 system is active. Next, the application typically allocates a statement handle and starts sending ODBC API calls to gather information on the capabilities of the database management system or to create and process SQL statements and stored procedure calls.

For further information on ODBC application structure and on the ODBC API, refer to *AS/400 Client/Server Performance Using the Windows Clients*, SG24-4526. You can also refer to the *ODBC SDK and Programmer's Reference*, available on the Microsoft Web site at: <http://www.microsoft.com/data/odbc>

11.4.4 ODBC configuration

The setup and configuration of ODBC Data Sources is discussed in “OneWorld and ODBC Data Sources” on page 191. You can also refer to the *OneWorld Tuning Guide* available at <http://www.as400.ibm.com/service/bms/jde-support.htm> for current recommendations for ODBC configuration.

11.5 License information

AS/400 Client Access Express for Windows (5769-XE1) is a licensed program product that is supported only on an AS/400 running OS/400 V4R2 or later. Two components require an AS/400 Client Access for Windows family (5769-XW1) license before you can use them:

- PC5250 Display and Printer Emulator
- Data Transfer

Note

If you are using a Client Access Express for Windows PC5250 emulator session in connection with Operations Console, you do not need a license for it even though the installation program of Client Access states that you do.

Other Client Access Express components do not require a license.

The type of installation you choose determines which components are installed.

The following list explains the types of installations you can choose and whether a component requires a license to support the installation:

- The Typical installation choice includes components that require a license.
- The PC5250 Display and Printer Emulation installation choice includes components that require a license.
- The Custom installation choice allows you to choose which components to install, including components that require a license. Components that require a license are identified as shown in Figure 115.
- The Full installation choice installs all components. A license is required.

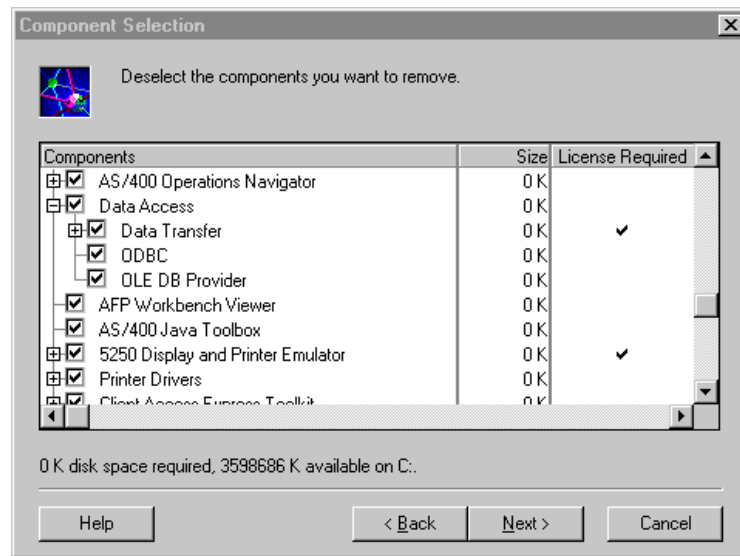


Figure 115. Component selection

To access all of the Operations Navigator functions discussed earlier, you must choose Custom or Full installation.

Client Access Express for Windows is licensed by the number of concurrently active PCs that are accessing AS/400 systems. PCs are counted when they connect to the AS/400 system and keep that license until they reach the license release timeout after they disconnect from the AS/400 system. When the license release timeout expires, the license is made available for another PC to use. If a PC connects to more than one AS/400 system, that PC requires a license on each AS/400 system to which it connects. A PC uses only one Client Access Express license per AS/400 system, regardless of the number of sessions that are established to that AS/400 system.

Keep in mind that ODBC connections do *not* require a license. For example, you can start many 5250 emulation, data transfer, or other sessions to the system, and the PC only requires one Client Access Express license.

Client Access Express licensing is done at the Client Access Express family level, not at the individual client-product level. Therefore, any combination of the Client Access Express products is allowed up to the limit of the license that was acquired. Customers who acquire a Client Access Express license are entitled to use the Client Access family products in any combination.

Chapter 12. Operations

As with any other application software, there are certain activities associated with the day-to-day operations of OneWorld on the AS/400 system. This chapter is beneficial to system administrators and other people interested in the management of the day-to-day operations and commands used to perform system maintenance.

Note

For the purposes of the examples in this chapter, the OneWorld release level is *B733*. If you are using a different release of OneWorld, substitute your actual release number, wherever the text *B733* occurs.

12.1 The OneWorld Installation Menu A98OWMNU

This section highlights the J.D. Edwards OneWorld Installation menu (Figure 116). This AS/400 menu is delivered in the JDEOW menu. To use this menu, you must have the JDEOW menu in your library list.

```
A98OWMNU                J.D. Edwards & Company                DENAS2
B733                    OneWorld Installation Menu

... DATA LIBRARIES
2.  Install Data Dictionary
3.  Install Central Objects

... SYSTEM LIBRARIES
4.  Install OneWorld Server Libraries
5.  Install Additional Path Codes

... PRINTER CONFIGURATION
7.  Create Output Queue
8.  Create Job Queue

... POST INSTALLATION
9.  Edit INI File
10. Verify Library List

... SERVER CONFIGURATION
12. Porttest
13. Start JDE Server
14. End JDE Server
15. Clear IPC

... MISC. OPERATIONS
17. Display Active Jobs
18. Work with Spooled Files

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
F13=Information Assistant  F16=AS/400 main menu
```

Figure 116. OneWorld Installation Menu

The following list provides a brief description of each menu option and what you can do with it:

- **Install Data Dictionary:** This option allows the user to install the OneWorld Data Dictionary on the AS/400 system. It should only be run once, since JD Edwards recommends only one Data Dictionary for all path codes.
- **Install Central Objects:** To install Central Objects, place the CD with the central objects in the CD-ROM drive before selecting this option. This option

defaults in the optical drive and base release level. The user is prompted to enter in the library that will hold the central objects, for example COPRDB733.

- **Install OneWorld Server Libraries:** The Enterprise Server installation process copies OneWorld system and path code files directly from the Enterprise Server CD. This option significantly reduces the time that was required in previous release to transfer the software from the Deployment Server.
- **Install Additional Path Codes:** This option installs additional path codes.
- **Create Output Queue:** This option prompts the user to create output queues that will handle reports and job logs.
- **Create Job Queue:** This option allows the user to create job queue.
- **Edit jde.ini File:** Allows you to edit the jde.ini file on the AS/400 system. The default release number is the current OneWorld release. Press Enter, and enter a 2 by the JDE entry listed.
- **Verify Library List:** This is a shortcut for the DSPLIBL command.
- **Port test:** This option invokes the program to initialize the user and an environment, after OneWorld was installed and configured correctly. This program verifies that OneWorld is installed correctly and must be run against every environment.
- **Start JDE Server:** This option starts JDE Server. If this is successful, you should see the entry NETWORK running with a PGM-JDENET_N in SELW status. There should also be an entry for SENTINEL running with a PGM-MONITOR in SELW status.
- **End JDE Server:** This option should be run before clearing IPC structures.
- **Clear IPC:** This option clears the IPC structures.
- **Display Active Jobs:** This option displays the current active jobs on the system by using the WRKACTJOB command.
- **Work with Spooled Files:** This option displays the reports and job logs in the output queue for the user by using the WRKSPLF command.

12.2 Setup tasks

This section describes some proactive operational tasks that can help your department meet its service level goals.

12.2.1 Activating Service Director

Contact your service provider to activate and configure Service Director on your AS/400 system. Service Director is an IBM-supplied application that monitors the status of your AS/400 hardware. If a hardware component reports an exception to OS/400, Service Director places a call to IBM hardware service personnel on the ECS communications line.

12.2.2 Creating a problem escalation contact list

Document the process of problem notification and resolution for your operational tasks. Include contact information for both the primary and an alternate person for each department served by your OneWorld environment. Record the

telephonic support contact procedures for each vendor application and operating system. Add this document to your operational binder.

12.2.3 Building a system information binder

Collect and maintain the following information for each OneWorld server:

- Machine model, type, and serial number
- System hardware configuration (on the AS/400 system, this is known as the “rack configuration”). This can be obtained in printed form via Systems Service Tools.
- A list of the installed applications with the version, service pack level, and vendor contact information
- TCP/IP configuration information
- System configuration information (system values, subsystem settings, communication configurations)
- A chronological log of hardware, service pack (PTF) installations, and system configuration changes

12.2.4 Determining key performance indicators

Key performance indicators are significant data points of system resource utilization. By monitoring and recording them on a regular basis, they can be used for input to the capacity planning and system health check status reporting processes. This information can be obtained through Performance Tools collections or commands such as Work with System Status and Work with Active Jobs. The best time to capture this information is the time of day and day of week when the server experiences the most usage. Typical indicators are:

- Percentage of utilized disk space
- Peak CPU utilization
- Peak response time
- Peak number of active jobs
- Total jobs on the system

12.2.5 Creating an operations schedule

Develop a calendar of significant events that may impact system availability for each server. Examples of such events are:

- Application periodic batch (UBE) processes (daily, weekly, monthly, etc.)
- Operational tasks (backups, purges, file reorganizations)
- System administrative tasks, such as hardware additions or removals, service pack (PTF) updates, software additions, removals, and system audits

12.3 Setting up multiple job queues for use by OneWorld

Configure OneWorld on the AS/400 system to use multiple job queues by performing following steps. The special authorities *SECADM and *ALLOBJ are required to complete these tasks.

1. Using the command Create job queue (CRTJOBQ), create job queues defined by major OneWorld product suites as follows:

- CRTJOBQ JOBQ (JDEFIN)
- CRTJOBQ JOBQ (JDEMFG)
- CRTJOBQ JOBQ (JDEHRM)
- CRTJOBQ JOBQ (JDETECH)
- CRTJOBQ JOBQ (JDEDST)

2. Use the ADDJOBQE command to add a job queue entry to an active subsystem. The CHGJOBQE command permits changes to the job queue's description, for example, maximum active jobs:

- ADDJOBQE SBS (QBATCH) JOBQ (JDEFIN) MAXACT (1) SEQNBR (11)
- ADDJOBQE SBS (QBATCH) JOBQ (JDEHRM) MAXACT (1) SEQNBR (12)
- ADDJOBQE SBS (QBATCH) JOBQ (JDEDST) MAXACT (1) SEQNBR (13)
- ADDJOBQE SBS (QBATCH) JOBQ (JDEMFG) MAXACT (1) SEQNBR (14)
- ADDJOBQE SBS (QBATCH) JOBQ (JDETECH) MAXACT (1) SEQNBR (15)

Note

J.D. Edwards strongly recommends that you use a single threaded job queue, which permits only one active job per subsystem per job queue. This ensures processing and data integrity as some OneWorld applications and processes must occur sequentially.

The need for this is clear when you consider this scenario: A user launches a UBE to post journal entries. A second user submits a report identifying recently posted journal entries by batch. If the job queue is multi-threaded, there is a good possibility that the second UBE completes prior to the first UBE that is responsible for the posting of journal entries. It is obvious the information contained in the second UBE's report would not provide the desired results.

3. Use the DSPSBS command to verify that the previously added job queues are recognized by the subsystem and were setup with the parameter MAX ACTIVE JOB of "1":

```
DSPSBS QBATCH
```

Select option 6, Display Job Queue Entries, to access the screen shown in Figure 117.

Display Job Queue Entries												
Subsystem description:		QBATCH	Status:		ACTIVE		System:				ASM23	
Seq	Job	Library	Max	-----Max by Priority-----								
Nbr	Queue		Active	1	2	3	4	5	6	7	8	9
10	QBATCH	QGPL	1	*	*	*	*	*	*	*	*	*
11	JDEFIN	QGPL	1	*	*	*	*	*	*	*	*	*
12	JDEHRM	QGPL	1	*	*	*	*	*	*	*	*	*
13	JDEDST	QGPL	1	*	*	*	*	*	*	*	*	*
14	JDEMFG	QGPL	1	*	*	*	*	*	*	*	*	*
15	JDETECH	QGPL	1	*	*	*	*	*	*	*	*	*
20	QS36EVOKE	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*
50	QTXTSRCH	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*

Figure 117. Display Job Queue Entries

4. Use the Create Job Descriptions (CRTJOB) command again defined by major OneWorld product suites, for example, JDEFIN, JDEHRM, JDEDST, JDEMFG, JDETECH:

- CRTJOB JOB (JDEFIN) JOBQ (JDEFIN) TEXT ('JDEFIN JOB')
- CRTJOB JOB (JDEHRM) JOBQ (JDEHRM) TEXT ('JDEHRM JOB')
- CRTJOB JOB (JDEDST) JOBQ (JDEDST) TEXT ('JDEDST JOB')
- CRTJOB JOB (JDEMFG) JOBQ (JDEMFG) TEXT ('JDEMFG JOB')
- CRTJOB JOB (JDETECH) JOBQ (JDETECH) TEXT ('JDETECH JOB')

5. If OneWorld Security Server is used, consider creating the following user profiles. Create user profiles defined by major OneWorld product suites, for example, JDEFIN, JDEHRM, JDEDST, JDEMFG, and JDETECH:

- CRTUSRPRF USRPRF (JDEFIN) JOB (*LIBL/JDEFIN) GRPPRF (ONEWORLD) OWNER (*GRPPRF)
- CRTUSRPRF USRPRF (JDEHRM) JOB (*LIBL/JDEHRM) GRPPRF (ONEWORLD) OWNER (*GRPPRF)
- CRTUSRPRF USRPRF (JDEDST) JOB (*LIBL/JDEDST) GRPPRF (ONEWORLD) OWNER (*GRPPRF)
- CRTUSRPRF USRPRF (JDEMFG) JOB (*LIBL/JDEMFG) GRPPRF (ONEWORLD) OWNER (*GRPPRF)
- CRTUSRPRF USRPRF (JDETECH) JOB (*LIBL/JDETECH) GRPPRF (ONEWORLD) OWNER (*GRPPRF)

6. From OneWorld menu GH9052, select **User Security**. Select each of your users and associate their “system user” with one of the previously defined product suites. For example, user BRENDA in the distribution department would be associated with “system user” JDEDST.

If you are not using OneWorld Security Server, it is necessary to update each OS/400 user profile's job description to point to the appropriate JOBQ. Therefore, user BRENDA's job description would be updated accordingly:

```
CHGJOB JOB (BRENDA) JOBQ (JDEDST)
```

Verify that the user profile is pointing to the correct job description and not to QDFTJOB. To do so, use the DSPUSRPRF BRENDA command. You should see a screen like the example in Figure 118 on page 304.

```

                                Display User Profile - Basic

User profile . . . . . : BRENDA

Initial program . . . . . : J98INITA
  Library . . . . . : SECLIB
Initial menu . . . . . : MAIN
  Library . . . . . : *LIBL
Limit capabilities . . . . . : *NO
Text . . . . . : Brenda Doe - Distri

Display sign-on information . . . . . : *SYSVAL
Limit device sessions . . . . . : *SYSVAL
Keyboard buffering . . . . . : *SYSVAL
Maximum storage allowed . . . . . : *NOMAX
  Storage used . . . . . : 00604
Highest scheduling priority . . . . . : 0
Job description . . . . . : BRENDA
  Library . . . . . : QGPL
Accounting code . . . . . :
Message queue . . . . . : BRENDA
  Library . . . . . : QUSRSYS

```

Figure 118. DSPUSRPRF (BRENDA)

In OneWorld versions *prior* to B733, OneWorld jobs submitted to the AS/400 system are defined entirely on OS/400-defined job queue concepts. These include the use of AS/400 user profile and job description characteristics that ultimately define a job queue. The job queue value is retrieved from the job description associated to the AS/400 user profile that launched the job. This differs from other OneWorld supported platforms that rely entirely on the use of the jde.ini file and OneWorld service scripts to define and select valid OneWorld job queues.

In OneWorld version B733, it is possible to assign a user's UBE job queue in the client workstation's jde.ini file for jobs that are run on the AS/400 platform. It is also possible to use OS/400 defined job queue concepts. Again, these include the use of AS/400 user profile and job description characteristics to define a job queue.

In the following example, the OneWorld client workstation's jde.ini file defines the job queue used (QJDEDST) when launching a UBE against an AS/400 platform:

```

[NETWORK QUEUE SETTINGS]
UBEQueue=QJDEDST
UBEPriority=5
PrintImmediate=FALSE
SaveOutput=TRUE
InstallSpecs=Y
JDENETTimeout=60

```

Here, the *UBEQueue* statement identifies the job queue assigned to this OneWorld client workstation. This value must be defined for use by OneWorld services running on the logic server.

On the Deployment Server, jde.ini is located in the \oneworld\bnnn\client\misc\ directory, where *bnnn* is OneWorld release. On the client workstation, jde.ini is located in the \Winnt\JDE.INI directory.

In the next example, the OneWorld client workstation's jde.ini file defines a null value instead of the job queue name. This causes OneWorld services to use OS/400 defined job queue concepts. The job queue value is then retrieved from the job description associated to the AS/400 user profile that launched the UBE job. If you are using OneWorld Security Server, the AS/400 "system" user profile's job description, defined for the OneWorld user, is used to define the job queue:

```
[NETWORK QUEUE SETTINGS]
UBEQueue=
UBEPriority=5
PrintImmediate=FALSE
SaveOutput=TRUE
InstallSpecs=Y
JDENETTimeout=60
```

A null value (blank) for UBEQueue forces OneWorld to use OS/400 defined job queue concepts. Therefore, the AS/400 user profile's job description JOBQ value is used to assign the job queue.

Note

As previously explained, the job queue used when submitting a OneWorld UBE job is defined by either the client workstation's jde.ini file or the AS/400 user profile's job description. However, the job queue assigned to a OneWorld AS/400 server package job is determined solely by the AS/400 user profile's job description used to launch this job.

12.4 Creating a command to set prestart job counts to greater than 1,000

A number of prestart jobs greater than 1,000 may be required if there is a large number of users that each require multiple ODBC sessions. Each ODBC session uses one copy of QZDASOINIT in the subsystem QSERVER.

A PTF that supports the API to change QSERVER's subsystem description information is available that can be used to set the initial value to a number larger than 1,000:

- For V4R1, the PTF number is SF48069.
- For V4R2, the PTF number is SF48071.
- For V4R3 and later, this API is in the base operating system.

12.5 Periodic tasks

This section explains how to perform some typical periodic operational procedures in OneWorld environment.

12.5.1 Replicating TAM specifications on the AS/400 system

It may become necessary to replicate Table Access Management (TAM) specifications (Specs) on the AS/400 system from one pathcode location to another. The following procedure identifies an AS/400 backup and recovery process used for accomplishing this task.

12.5.1.1 Eliminating all locks to the PRODB733 TAM Specs

Perform the following process:

1. Eliminate all locks to the PRODB733 TAM specifications to ensure that these link objects are saved in the appropriate state. This can be accomplished by ending OneWorld services and clearing inter-process communication (IPC) locks. Issue the following commands while signed on as user ONEWORLD:

```
ENDNET  
CLRIPC
```

2. Sign off and sign on as a user with administrative capabilities. End all ONEWORLD user jobs. This further ensures that the PRODB733 TAM specifications are free of IPC locks (in QTEMP) held by users having run Porttest or RUNUBE:

```
WRKUSRJOB ONEWORLD *ACTIVE
```

Enter a 4 in front of each of the jobs that are displayed. On the CL command line, type:

```
OPTION(*IMMED)
```

Press Enter twice.

12.5.1.2 Saving the IFS link directory and the objects within it

Complete the following tasks:

1. Create a save file on the AS/400 system:

```
CRTSAVF QGPL/LINKPROD
```

2. Save the IFS PRODB733 directory and the link objects contained within it:

```
SAV DEV('QSYS.LIB/QGPL.LIB/LINKPROD.FILE') OBJ('/PRODB733')  
OUTPUT(*PRINT) INFTYPE(*SUMMARY)
```

12.5.1.3 Restoring the IFS link directory and the objects within it

Follow this process:

1. Remove all link objects from the CRPB733 directory:

```
RMVLNK(CPRB733/specfile/*)  
RMVLNK(CPRB733/specfile)
```

2. Restore the IFS directory and link objects:

```
RST DEV('QSYS.LIB/QGPL.LIB/LINKPROD.FILE') OBJ('/CRPB733' *INCLUDE *SAME)  
OUTPUT(*PRINT) INFTYPE(*SUMMARY)
```

12.5.1.4 Deleting the CRPB733 TAM specification global tables

Remove the global table link objects from CRPB733 directory:

```
RMVLNK(CPRB732/specfile/gbltbl*)
```

These global table object links are recreated dynamically either through Porttest or the submission of a UBE against an environment associated to the CRPB733 pathcode.

Note

Deploy a server package (against pathcode CRPB733) containing those modified objects in the CPRB733 pathcode that do not exist in the PRODB733 pathcode.

12.5.2 Starting and stopping the OneWorld system

This section covers the best practices for starting and stopping OneWorld.

12.5.2.1 Starting OneWorld manually

OneWorld can be automatically started by modifying the AS/400 system startup program. To manually start OneWorld on the AS/400 system, perform the following steps:

1. Sign on to the AS/400 system as `ONEWORLD`.
2. Select the release of OneWorld (Figure 119). Possible environments depend on the release of OneWorld. A few environments from the B733 release include: PRD733, PRT733, CRP733, TST733, and DEV733.

```
Current OneWorld Versions

Type option, press enter.
  1=Select (BL)
O
P Version IPC  Port          Description
_ B733      2101  6006          B733 Release
```

Figure 119. OneWorld Signon: Select the OneWorld release

3. Start JDENet using the following command:

```
STRNET
```

4. If you need to verify that the basic OneWorld installation is operational, run the Porttest command:

```
PORTTEST JDEUSR JDEPWD CRP733
```

Here, *JDEUSR* represents the OneWorld AS/400 user ID, *JDEPWD* represents the password, and *CRP733* represents the OneWorld environment.

For further detailed information, refer to 22.4, “Installation verification using PORTTEST” on page 755.

5. Use the `WRKACTJOB` command to verify that OneWorld is running. Type the following command:

```
WRKACTJOB SBS(JDEB733)
```

Verify the entry `NETWORK` with function `PGM_JDENET_K` and that the status of `SELW` is running (until a “net request” is performed, the CPU is “0”). The `STRNET` command also starts a second job called `SENTINEL`. See Figure 120.

```

                                Work with Active Jobs
                                ASM23
                                03/01/99 11:23:20
CPU %:      .0      Elapsed time: 00:00:00      Active jobs: 177
Opt Subsystem/Job  User      Type  CPU %  Function      Status
      JDEB733      QSYS      SBS    .0      DEQW
      JDENET_K      ONEWORLD  BCI    .0      DEQW
      JDENET_K      ONEWORLD  BCI    .0      DEQW
      JDENET_K      ONEWORLD  BCI    .0      DEQW
      JDENET_K      ONEWORLD  BCI    .0      DEQW
      JDENET_K      ONEWORLD  BCI    .0      DEQW
      NETWORK      ONEWORLD  BCH    .0      PGM-JDENET_N  SELW
      SENTINEL     ONEWORLD  ASJ    .0      PGM-MONITOR   SIGW

====>
F21=Display instructions/keys

                                Bottom

```

Figure 120. Active subsystem JDEB733

12.5.2.2 Stopping OneWorld

Before stopping OneWorld, ensure that your library list is set correctly. The `DSPLIBL` command displays the current library list. Verify that `B733SYS` is in the library list. End JDENet by using the following command:

```
ENDNET
```

The End OneWorld Services (`ENDNET`) command is in the system library, for example, `B733SYS`. This command instructs OneWorld to end the JDENet jobs and clean up all runtime structures.

It may be necessary to clean up the OneWorld environment. Read the following section to learn more about handling abnormal termination of OneWorld.

12.5.2.3 Recommended AS/400 system shutdown

To ensure an orderly shutdown and *rapid* subsequent IPL, we recommend that you use the following procedure when shutting down the system.

Shutting the system down with `PWRDWN SYS *IMMED` command works, too. However, on the subsequent IPL, the system will spend extra time (in some cases, a significant amount of time) cleaning up jobs that were abnormally terminated.

The following procedure assumes that job logs are not required. If they are, either individual jobs should be ended or the `*NOJOBLOG` option should *not* be used on these commands.

1. All OneWorld users should sign off.
2. Hold all batch job queues. Wait for all batch jobs to complete.
3. Issue the `ENDNET` command to shutdown OneWorld subsystem.
4. End the QSERVER subsystem by using the following command:


```
ENDSBS SBS(QSERVER) OPTION(*CNTLD) DELAY(*NOLIMIT)
ENDSBSOPT(*NOJOBLOG)
```

- To see which USERS are still connected or had connections that were “broken”, enter the `NETSTAT *CNN` command to get the IP addresses. It will be the sessions with a local port of “as-data” as shown in Figure 121.

```

Work with TCP/IP Connection Status
System: AS29
Local internet address .....: *ALL

Type options,press Enter.
4=End 5=Display details

Remote Remote Local
Opt Address Port Port
- * * as-cent> 083:50:31 Listen
- 9.5.33.114 3965 as-data> 000:00:01 Established <==
- * * as-dtaq-s 083:50:32 Listen

```

Figure 121. Netstat

- Force ending of any remaining QSERVER jobs by entering the following command:

```
ENDSBS SBS(QSERVER) OPTION(*IMMED) ENDSBSOPT(*NOJOBLOG)
```

- Shutdown any Integrated Netfinity Servers by ending any running applications, logging off and then issuing the `VRYPFF` command for the associated *NWS interface.
- Wait for the QSERVER subsystem to end and any Integrated Netfinity Adapters to be varied off.
- End all other user subsystems with the `ENDSBS` command.
- End the system jobs to ensure that journaling, commitment control, and anything else completes. Enter the following command:

```
ENDSYS OPTION(*CNTLD) DELAY(*NOLIMIT) ENDSBSOPT(*NOJOBLOG)
```

- Once all the subsystems have ended, except for QSYSWRK, and the controlling subsystem and the jobs in these two subsystems have ended, issue the following command:

```
ENDSYS SBS(QSERVER) OPTION(*IMMED) ENDSBSOPT(*NOJOBLOG)
```

This forces QSYSWRK or any other subsystems that have not ended to end (with the exception of the controlling subsystem).

- When subsystem QSYSWRK has ended, power down the system with the command:

```
PWRDWN SYS OPTION(*CTLD) DELAY(300) ENDSBSOPT(*NOJOBLOG)
```

12.5.3 Daily health check tasks

Perform the following set of tasks at the beginning of each operations shift:

- Check system error logs using System Service Tools (SST) for any significant hardware or software errors.

2. Review system operator messages (command `DSPMSG QSYSOPR`) for any significant exceptions or messages requiring a response.
3. Review the history log for any significant exceptions by using the `DSPLOG` command.
4. Check disk and CPU utilization by using the Work With System Status (`WRKSYSSTS`) command.
5. Check the system panel for attention light status and IPL state.
6. If tape operations were scheduled to run in the previous shift, check their current status.
7. Using the Work With TCP/IP Network Status (`NETSTAT`) command, ensure that the appropriate TCP/IP interfaces are active.
8. Verify that all appropriate communication lines and Network Servers are active.
9. Verify that OneWorld is in an active state by checking for the presence of sentinel and network jobs within the `JDEBxxxx` (where `xxxx` represents the OneWorld release) subsystem.

12.5.4 System management with Operations Navigator and Management Central

With Client Access Express, the AS/400 system offers some powerful GUI-based system management tools. Operations Navigator with Management Central makes the tasks of systems management much easier. If you are new to the AS/400 system, you may find that the GUI view of your system is easier for you to use. If you are familiar with the original AS/400 5250 interface (green screen), you may appreciate the added functionality and ease of use.

See 11.2, “Operations Navigator and Management Central” on page 286, for an overview of Operations Navigator and Management Central functions.

This section provides some hints on how to use them to perform system management tasks. For additional examples and more information, visit these Web sites:

- Operations Navigator: http://www.as400.ibm.com/oper_nav
- Management Central: <http://www.as400.ibm.com/sftsol/MgmtCentral.htm>

Note

Management Central can be CPU intensive when running some of its tasks, such as data gathering. It is set up to run at a run priority of 15. It can also utilize disk space on your central machine if it is set up to capture performance data from your other systems. Make sure that you understand its capabilities and uses as you implement it. Ensure that it is running on only one PC at a time.

Figure 122 shows the Management Central and Operations Navigator screen that you see when you start the Operations Navigator.

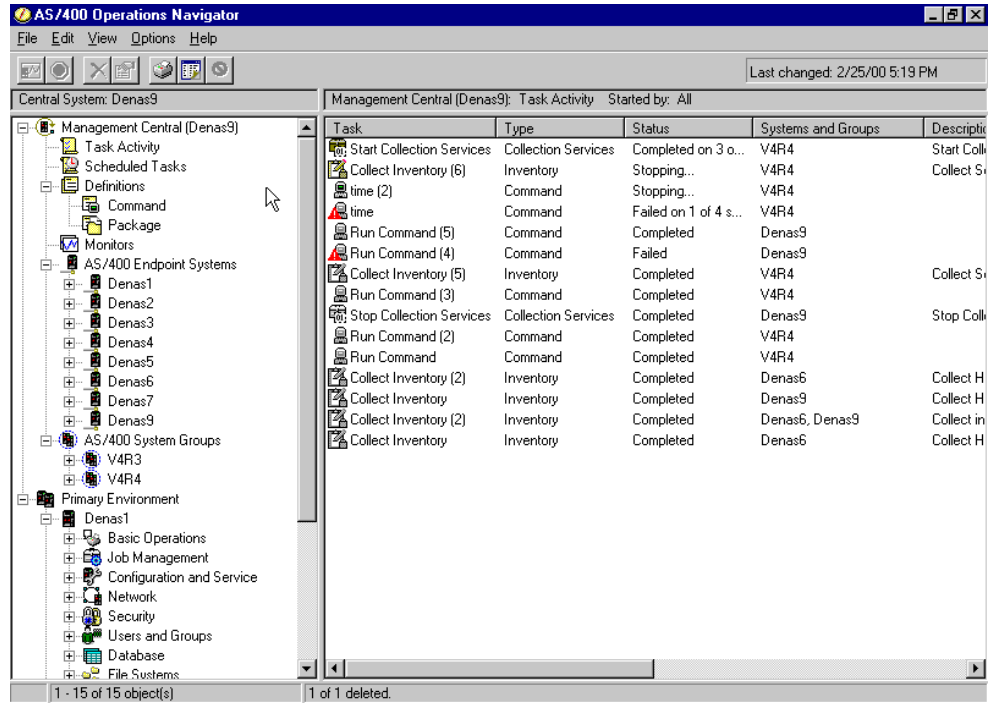


Figure 122. Management Central main panel

For more information on Management Central, refer to *Management Central: A Smart Way To Manage AS/400 System*, SG24-5407.

12.5.4.1 Working with spooled files

To view, manage, and print your AS/400 spool files via GUI performing following steps:

1. Start Operations Navigator.
2. Double-click the icon with your system name.
3. Double-click **Basic Operations**.
4. Double-click **Printed Output**.

To delete a spooled file, right-click the spooled file, and then click **Delete** or press the Delete key (Figure 123 on page 312).

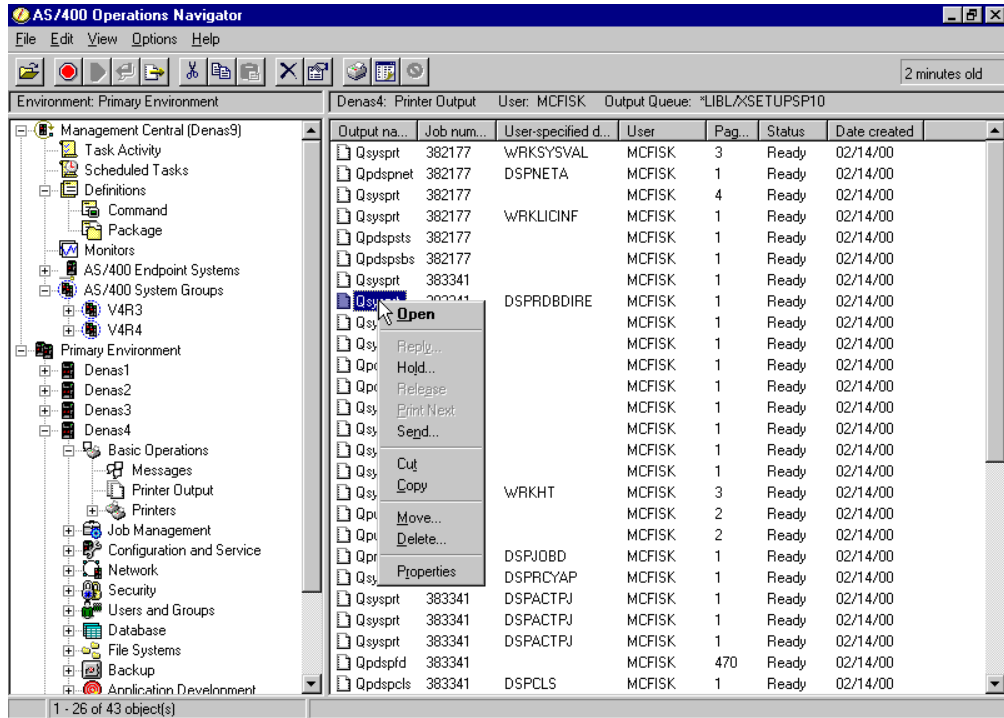


Figure 123. Operations Navigator spooled output

To delete all spooled files, follow these steps:

1. Click the first spooled file.
2. Hold down the Shift + End keys.
3. Press the Delete key.
4. Click **Delete** to confirm. You may deselect certain files before you delete them. You can also delete the chosen spooled files using your mouse.

To display or print the spooled file, you can drag and drop your spooled files to your desktop or onto a diskette (or to a different AS/400 system via TCP/IP). After that, you can double-click it, and your PC editor (such as Notepad) will display or print it.

12.5.4.2 Working with messages

From an Operations Navigator display, you can view, send, and delete messages from a user's message queue or even QSYSOPR message queue. To do this, perform following steps:

1. Within Operations Navigator, expand your AS/400 system icon.
2. Sign on to the AS/400 system.
3. Expand **Basic Operations**.
4. Double-click **Messages**. At this point, you can pull down the Options tab and click **Include** to access the system operator's message queue.

12.5.4.3 Working with OneWorld files

Through the NetServer function of Client Access Express, you can work with OneWorld files that are stored in the integrated file system. You can access them

either through Operations Navigator or Windows Explorer. You can also move, copy, delete, or edit these files. To work with them through Operations Navigator (see Figure 124), complete these tasks:

1. Expand your system.
2. Expand **File systems**.
3. Expand **Integrated File System**.
4. Search for your directory and files.

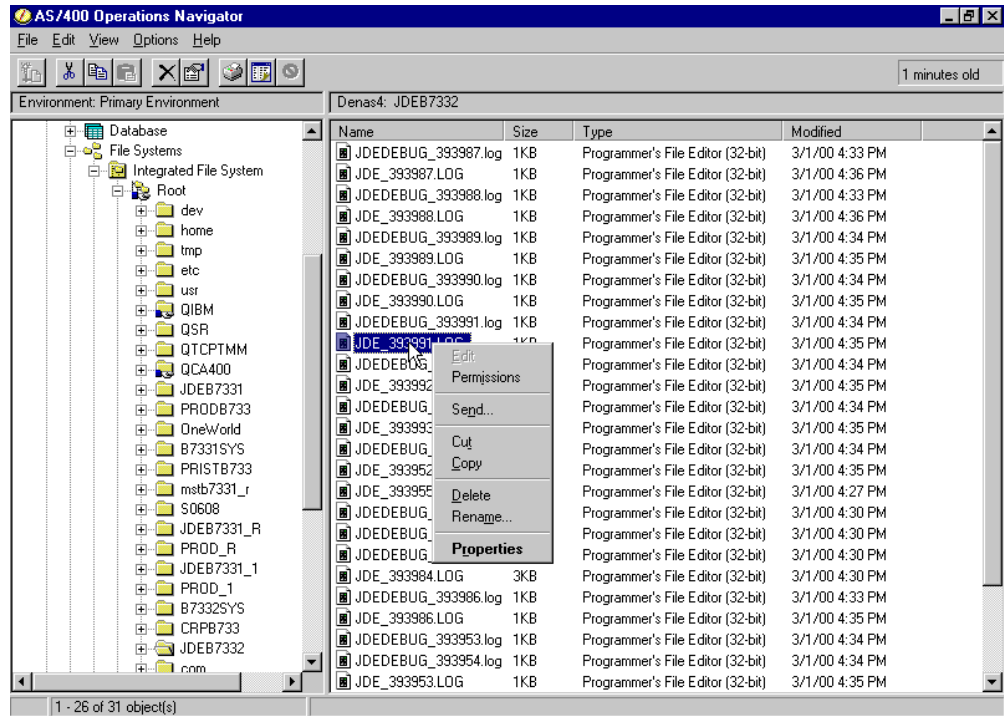


Figure 124. Handling IFS files

12.5.4.4 Running SQL commands

From Operations Navigator, you can run interactive SQL commands on the AS/400 server, from a GUI and without a need for additional licensed program on the AS/400 system. In Operations Navigator, right-click **Database**. Then select **Run SQL scripts** and follow the instructions (Figure 125 on page 314).

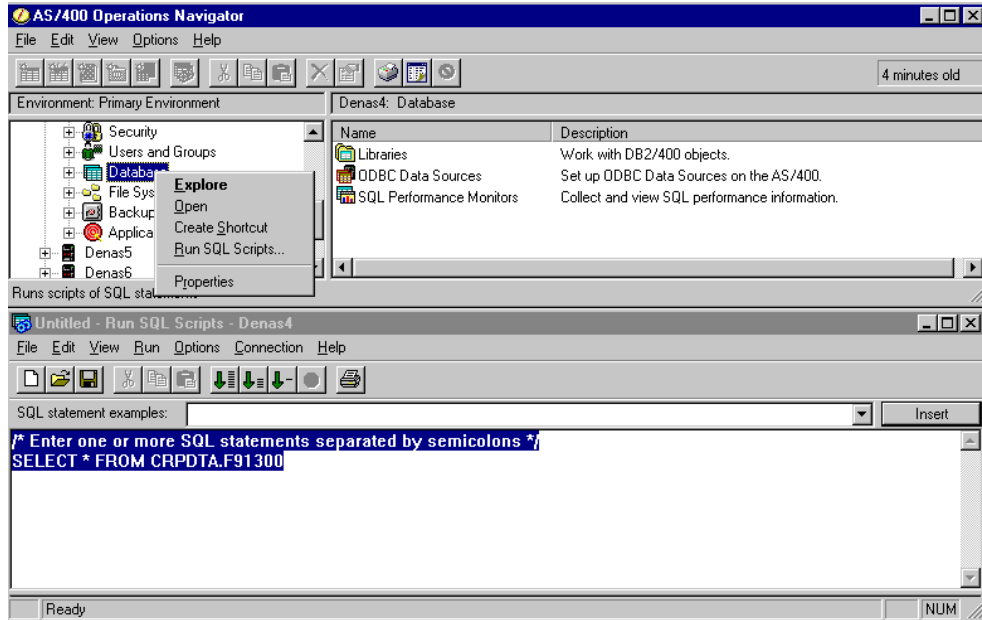


Figure 125. Run SQL scripts

12.5.4.5 System monitoring

Through Management Central, you can establish a monitoring session for either a single AS/400 system or multiple AS/400 systems to check for a variety of measurements such as AS/400 disk or CPU utilization. You can set a threshold for each parameter. When it is reached, Management Central will highlight the graph line that triggered the event in red. You can also have Management Central sound an alarm or submit a (client or host) command when the threshold is reached. As shown in Figure 126, you can review disk percent full and set a threshold of 80%. While the threshold's set point is indicated in red along the left margin, the title bar is also in red to indicate a threshold has been reached. The systems above 80% are highlighted in red. As you place the cursor on one of the graph points, it opens a box showing you the time, the % full, and the name of the system. Similarly, when tracking CPU utilization, you can zoom in on your busiest machine, right-click on the end point, and select Show details, which opens the detailed graph in the upper right square. If you right-click again to show the properties on a job, it opens the bottom right box of details.

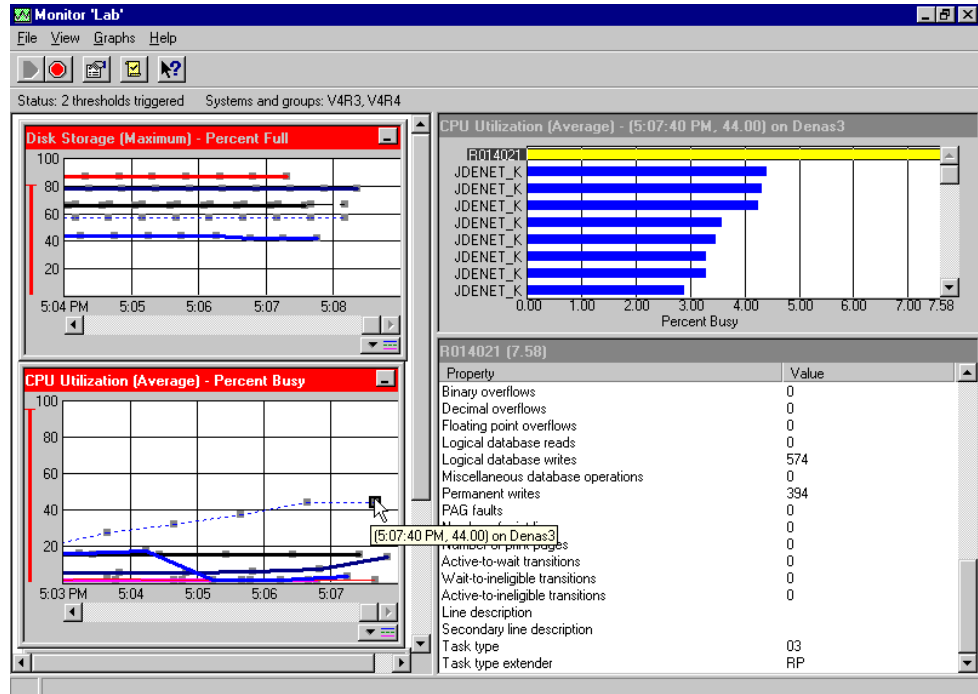


Figure 126. Management Central monitor

For a different view of your thresholds, you can also pull down the File tab and then Event Log tab to see the historical detail concerning when your thresholds were reached (Figure 127).

Event	Date	Time	System	Metric
Trigger	2/25/00	2:27:08 PM	Denas1	CPU Utilization (Average)
Trigger	2/25/00	2:23:36 PM	Denas9	CPU Utilization (Average)
Trigger	2/25/00	1:27:29 PM	Denas3	CPU Utilization (Average)
Trigger	2/25/00	11:45:17 AM	Denas1	Disk Storage (Maximum)
Trigger	2/25/00	11:45:10 AM	Denas2	Disk Storage (Maximum)
Trigger	2/25/00	11:23:53 AM	Denas1	Disk Storage (Maximum)
Trigger	2/25/00	11:23:46 AM	Denas2	Disk Storage (Maximum)
Trigger	2/25/00	10:58:49 AM	Denas2	Disk Storage (Maximum)
Trigger	2/25/00	10:58:45 AM	Denas1	Disk Storage (Maximum)

Figure 127. Monitor thresholds reached via Event Log

Additionally, you can also pull down the File tab and then the Status tab to see an overview of how many thresholds you have reached. See Figure 128 on page 316.

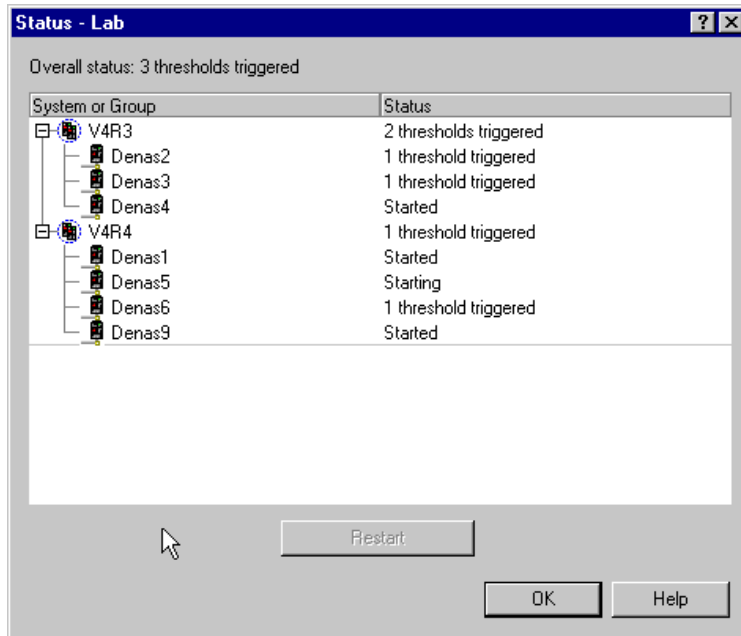


Figure 128. Monitor thresholds reached via the Status display

12.5.4.6 Checking the server status

You can also check the status of the various AS/400 TCP/IP server and Client Access Express server jobs. For example, you can drill down through Network->Servers->TCP/IP to see if your host servers are started (see Figure 129). Similarly, to make sure that your ODBC connections are active, you can view your Client Access servers, by drilling down through Network->Servers->Client Access (Figure 130). Your ODBC connections are made by the Database Server in that list. At this point, you can right-click your QZDASOINIT job and perform the functions in the pull-down as indicated in Figure 131 on page 318.

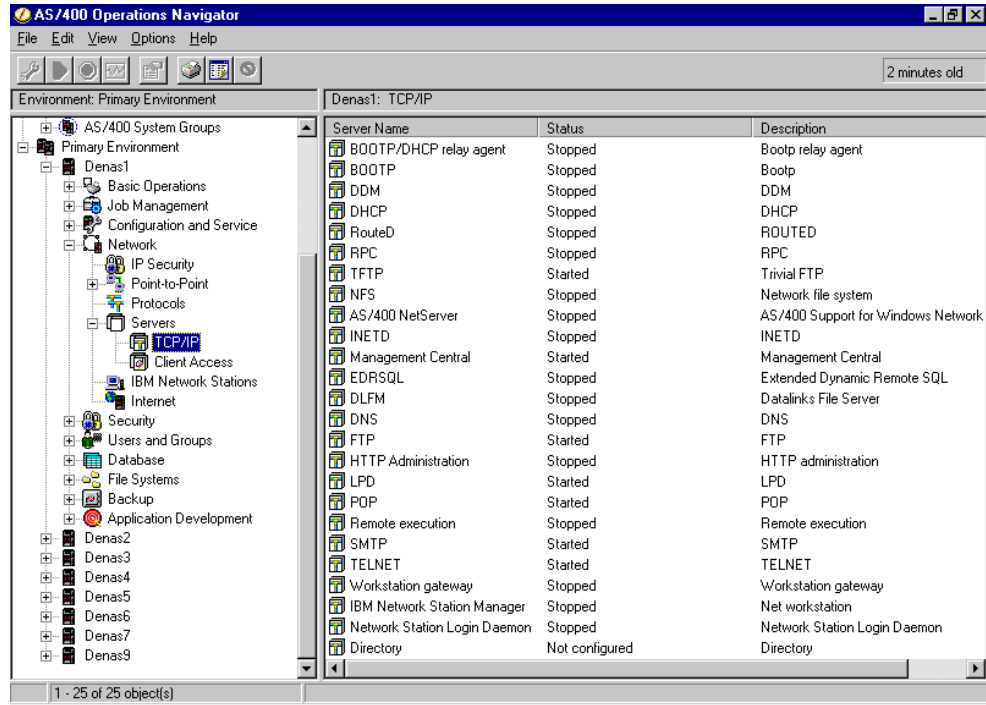


Figure 129. TCP/IP servers

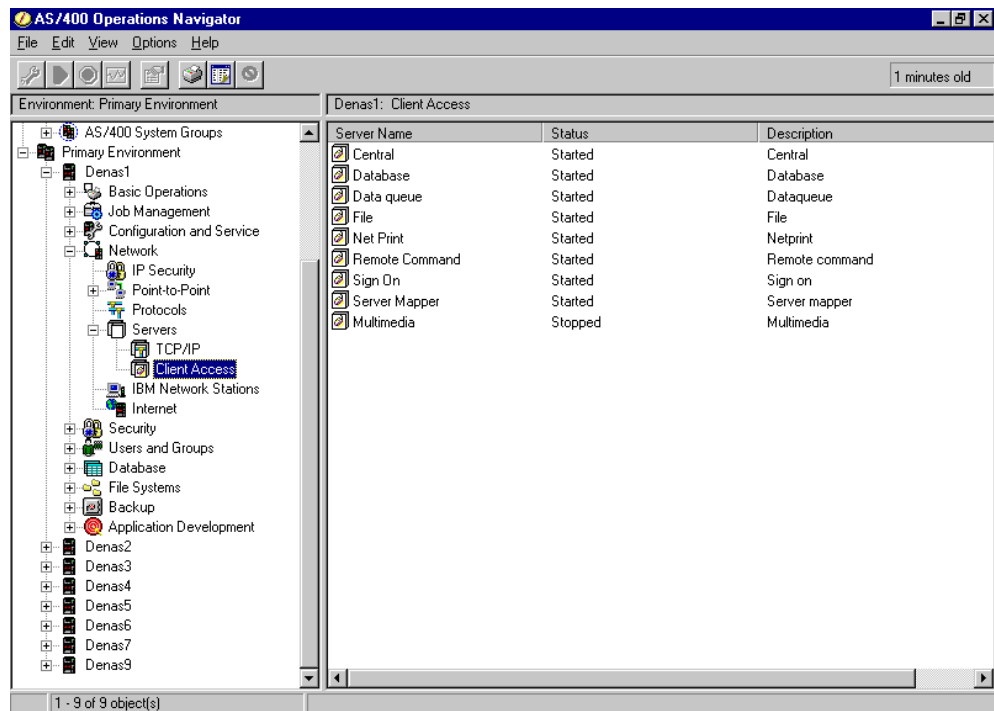


Figure 130. Client Access servers

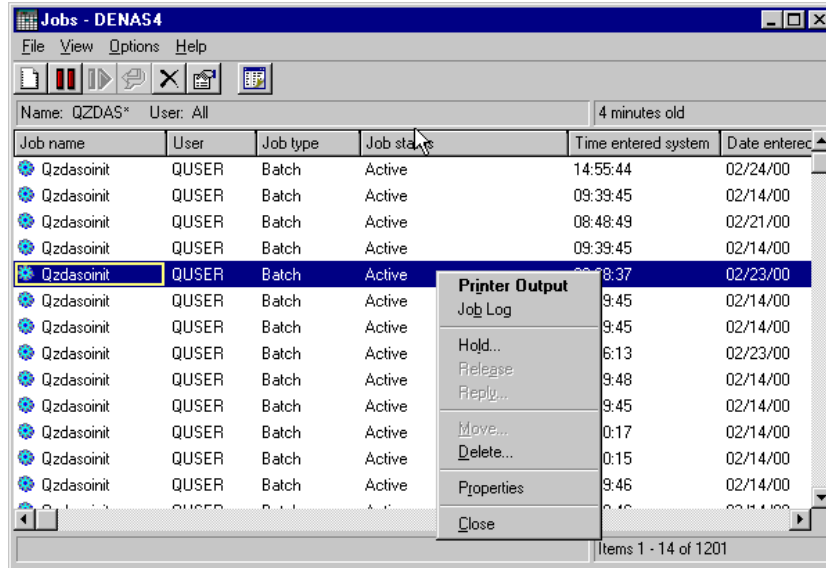


Figure 131. Viewing/managing QZDASOINIT jobs

12.5.4.7 Managing PTFs

Whether you have one AS/400 system or multiple ones, you can use Management Central to review your list of program temporary fixes (PTFs or fixes) and maintain its currency. With multiple AS/400 systems, you can set up one system as a model AS/400 system, and then keep the PTFs on the other machines current with the model AS/400 automatically through Management Central.

Within Management Central, expand either **AS/400 Endpoint Systems** or **AS/400 System Groups**, right-click the system or group, and select **Fixes**. Then, select **Compare and Update**. You can either view the differences or send and install the missing fixes on the target system (OS/400 V4R4 or higher is required).

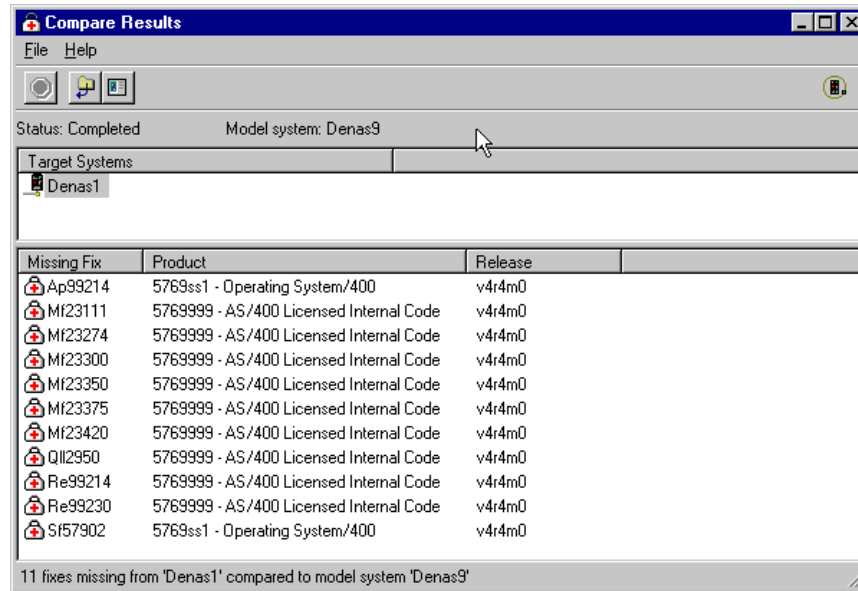


Figure 132. Comparing PTF levels between AS/400 systems

12.5.4.8 Remote commands

With Management Central, you can submit a remote command to all of your AS/400 systems simultaneously, either immediately or as a scheduled task. So if you have repetitive, periodic commands to run, you can set them up by expanding **Management Central->Definitions**, right-clicking **Command**, and then clicking **New Definition**. You can click **Previous Commands** to select from a list of those commands that were previously run or you can click **Check Syntax** to have the central system check your command for validity.

If you enter the Options tab, you can also decide how to set job log options (for example, always print a job log or only if the job fails) and whether to automatically reply to messages with the default or the system reply list. You can then start your command immediately, or you can schedule it (Figure 133 on page 320).

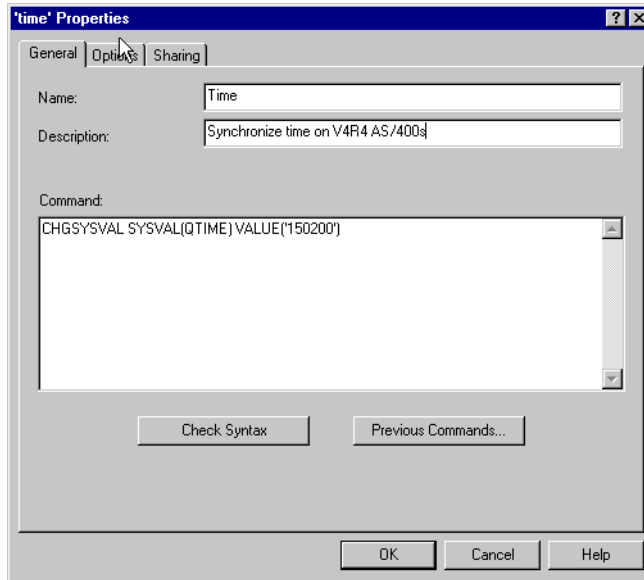


Figure 133. Submitting a remote command from Management Central

12.6 Cleaning up the OneWorld environment

If OneWorld ends abnormally, you may need to manually perform clean up tasks on the AS/400 Enterprise Server. It is possible that Interprocess Communication (IPC) structures may not be cleaned up following an execution of the ENDNET command. This may cause further problems when trying to start JDENet. If the IPC structures are not properly removed by the ENDNET command, you may manually remove them. IPC structures may become locked by an interactive job. For example, you may have to sign off and sign on to perform a successful clean up.

The OneWorld AS/400 server is shipped with the DSPIPC and CLRIPC commands, which allow you to display the IPC-related information and to remove IPC structures.

Note

If you find that the IPC locks do not clear, it is possible that another interactive job or batch UBE is running. These object locks must be resolved before continuing.

If tracing is turned on, in addition to IPC, you should clear the jde.log and the jddebug files. This keeps the files from becoming too large and clears out old messages.

Note

Do not clear IPC structures unless you are ready to restart JDENet process.

To clean up the OneWorld environment, perform the following steps:

1. Ensure that your library list is correct before attempting the clean up procedure.
2. Enter the following IPC commands:

```
DSPIPC
CLRIPC
```

The DSPIPC command produces the screen shown in Figure 134.

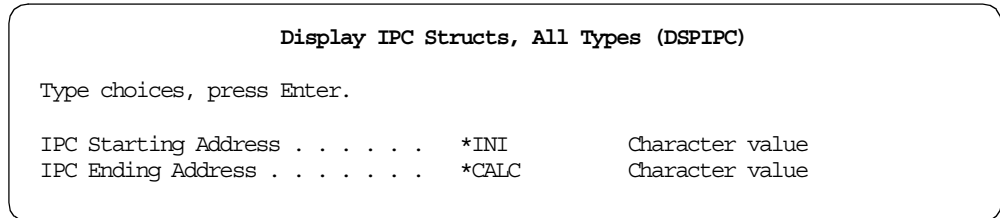


Figure 134. Display IPC (DSPIPC)

Use the “from” and “to” parameters to specify the starting and ending IPC addresses on which you want to operate. The default for the “from” parameter is *INI. This is the address specified in the INI file. The default for the “to” parameter is *CALC. This means that the value is calculated based on the value of the “from” parameter.

Note

IBM OptiConnect and OptiMover products use the IPC shared memory address 9999. Avoid setting the jde.ini file IPCStartKey setting to a starting value using the range of 9000 to 9999.

3. Clear the jde.log files:

```
DEL '/jde733/jde_*
```

4. Clear the jdedebug files:

```
DEL '/jde733/jdedebug*'
```

12.7 Administrating the batch processes on the AS/400 system

Administrating batch processes implies knowing what processes run when OneWorld starts, where files are placed before and after printing, and how to monitor those processes. Depending on how the software is installed, jobs run under several subsystems on the AS/400 system. The first subsystem, JDEB733, is created during the installation process and is responsible for running the OneWorld network and kernel processes. QBATCH is the default subsystem that jobs run in, but you can use other subsystems to distribute the workload.

12.7.1 Monitoring batch processes

To monitor batch processes, log on to the AS/400 Enterprise Server using an administrative user profile. Enter the following AS/400 command:

```
WRKACTJOB SBS (subsystem)
```

Specify the AS/400 subsystem that is appropriate, for example:

```
WRKACTJOB SBS(JDEB733)
```

```
Work with Active Jobs                                ASM23
                                                    03/01/99 09:33:45
CPU %:      .0      Elapsed time:  00:00:00      Active jobs:  181
Opt Subsystem/Job  User      Type  CPU %  Function  Status
-----
JDEB733          QSYS      SBS    .0
JDENET_K         ONEWORLD  BCI    .0
JDENET_K         ONEWORLD  BCI    .0
JDENET_K         ONEWORLD  BCI    .0
JDENET_K         ONEWORLD  BCI    .0
JDENET_K         ONEWORLD  BCI    .0
NETWORK         ONEWORLD  BCH    .0  PGM-JDENET_N  SELW
NETWORK         ONEWORLD  BCI    .0
SENTINEL        ONEWORLD  ASJ    .0  PGM-MONITOR   SIGW

====>
F21=Display instructions/keys

Bottom
```

Figure 135. WRKACTJOB SBS(JDEB733)

The output shown in Figure 135 shows that two NETWORK processes are running, but are currently waiting.

The NETWORK jobs are responsible for accepting and queuing the request on the AS/400 for processing, while the QBATCH subsystem is responsible for the execution of the report. To monitor the batch requests, specify:

```
WRKACTJOB SBS(QBATCH)
```

```
Work with Active Jobs                                ASM23
                                                    03/01/99 14:34:09
CPU %:      1.4      Elapsed time:  00:46:55      Active jobs:  193
Opt Subsystem/Job  User      Type  CPU %  Function  Status
-----
QBATCH          QSYS      SBS    .0
R04424          JDE       BCH    .0  PGM-PRINTUBE  RUN

====>
F21=Display instructions/keys

Bottom
```

Figure 136. WRKACTJOB SBS(QBATCH) where a batch job is active

As shown in Figure 136, the R04424 job is the report running at this time. The program, PRINTUBE, is the job responsible for running and printing the request. When the job is finished, it leaves the queue. The print job is either printed and deleted or saved as a member of file B733SYS/PRINTQUEUE.

12.8 Reviewing batch output files

If users submit a batch report to run on the AS/400 system, a corresponding output file is created on the Enterprise Server. It can be viewed on the user's workstation or printed by the server.

The default location for storing a Portable Document Format (PDF) file is based on values established in the OneWorld jde.ini file. This library name is normally B733SYS. However, it depends on how OneWorld was installed and whether the site installation has overridden the default setting.

This system library contains a PRINTQUEUE file. This file is the location for the PDF report requests. Members are added to this file each time a PDF report is completed.

To see a list of PDF files on the AS/400 system, perform these tasks:

1. Log on to the AS/400 Enterprise Server using an administrative user profile.
2. Enter the following command:

```
WRKMBRPDM B733SYS/PRINTQUEUE
```

```
Work with Members Using PDM

File . . . . . PRINTQUEUE
Library . . . . . B733SYS          Position to . . . . .

Type options, press Enter.
2=Edit      3=Copy  4=Delete 5=Display  6=Print  7=Rename
8=Display description  9=Save 13=Change text 14=Compile 15=Create module...

Opt Member   Date       Opt Member   Date       Opt Member   Date
F19          02/19/99   F34          02/22/99   F42          02/22/99
F26          02/22/99   F35          02/22/99   F43          02/22/99
F27          02/22/99   F36          02/22/99   F44          02/22/99
F29          02/22/99   F37          02/22/99   F45          02/22/99
F30          02/22/99   F38          02/22/99   F46          02/23/99
F31          02/22/99   F39          02/22/99   F47          02/23/99
F32          02/22/99   F40          02/22/99   F48          02/23/99
F33          02/22/99   F41          02/22/99   F49          02/23/99
More...
```

Figure 137. List of all PDF files that exist on the Enterprise Server

The naming convention for each member is based upon the OneWorld job number, which is a unique number assigned when the report is submitted. This number is a unique print request ID and is incremented each time a report is submitted to the Enterprise Server, regardless of whether the job is successful or has failed. It is not related to the process ID or job number that the AS/400 system assigns the batch job.

J.D. Edwards has an application program, P986116 (Submitted Reports) to display job numbers from the end-user OneWorld workstation. Also the system administrator can use the Work With Servers application (P986116) to display job numbers. These applications use the Job Master Search form to display job numbers that correspond to member names. You can use either application to delete PDF files by deleting the appropriate entries.

Two other settings, based on the workstation's `jde.ini` file, tell the server whether to print the report immediately upon completion and to save the output from the report, or delete it:

```
[NETWORK QUEUE Settings]
SaveOutput=TRUE
PrintImmediate=TRUE
```

Setting `SaveOutput` to `TRUE` causes the Enterprise Server to save the PDF files in `B733SYS/PRINTQUEUE` until you explicitly delete them. Setting the `PrintImmediate` value to `TRUE` tells the Enterprise Server to print the job immediately after completion of the report.

Setting the `SaveOutput` value to `FALSE` is encouraged. Otherwise, the workstation user should periodically delete the entries using OneWorld Job Master Search applicable in the Submitted Reports application (P986116).

12.9 Encoding the passwords of users who submit batch jobs

On the AS/400 system, when you want to encode user passwords for batch jobs, you need to change settings in the `[SECURITY]` section in the `B733SYS/INI` file. To encode passwords, edit the file with the `Edit File (EDTF)` command and modify the following setting:

```
[SECURITY]
ServerPswdFile=True
```

To deactivate encoding, change this setting to `FALSE`.

12.10 Integrated Netfinity Server operations

There are certain activities associated with the operations of the Integrated Netfinity Server. This section discusses some of these activities. For more information, refer to about Integrated Netfinity Server, refer to Chapter 18, "The Integrated Netfinity Server and OneWorld" on page 513.

12.10.1 Checking the status of the Integrated Netfinity Server

To view the status of the Integrated Netfinity Server, enter the following command:

```
WRKCFGSTS *NWS
```

The display shown in Figure 138 appears.


```

Work with Configuration Status                                03/03/99 15:26:40
Position to . . . . . Starting characters

Type options, press Enter.
 1=Vary on   2=Vary off  5=Work with job  8=Work with description
 9=Display mode status 13=Work with APPN status...

Opt  Description      Status      -----Job-----
      NWSD              ACTIVE
      TRNLIN           ACTIVE
      TRNLINET00      ACTIVE
      TRNLITCP00     ACTIVE      QTCPIP      QTCP      004408

```

Figure 138. Integrated Netfinity Server status

12.10.2 Starting and stopping the Windows NT server

The Integrated Netfinity Server can be shut down by selecting option 2, Vary off, on the display shown in Figure 138 or by using the `VRYPCFG` command. The Windows NT server can also be ended by using the Shutdown option from the Start menu of the Integrated Netfinity Server console.

Note

Use caution when varying off the Network Server description since any line attached to it will also be varied off. Furthermore, depending on your configuration, TCP/IP services may be adversely affected.

Select option 1, Vary on, to start the Integrated Netfinity Server.

For more information, refer to on what is happening during the startup of Integrated Netfinity Server, see 18.5.1, “Starting the Windows NT Server” on page 524.

Chapter 13. OneWorld package management

This chapter describes the processes used to maintain, promote, and deploy custom modifications created with OneWorld Tools. It includes an explanation of such concepts as package management planning, package deployment methodology, custom modifications, defining and building a package, and server packages.

13.1 Package management planning

This section contains recommendations, suggestions, and considerations for promoting custom modifications. It is imperative that you devise and follow a reliable version control and promotion process.

13.1.1 Understanding packages

The purpose of a package is to group OneWorld software modifications so that they may be deployed to workstations. A package describes where to find the components that you intend to deploy to workstations. A package can contain everything needed to run OneWorld (as in an initial installation for a new workstation) or a mere update.

13.1.1.1 Package types

There are three package types: *full package*, *partial package*, and *update package*. Each has its own advantages, disadvantages, and intended purpose.

Full package

The full package includes all OneWorld applications. Using full packages has the following advantages and disadvantages:

- **Advantages**

- A full package includes a complete suite of OneWorld applications (all specifications and foundation).
- Developers have local access to all OneWorld objects.
- After initial client workstation installation, network traffic is lower than with a partial workstation configuration because all applications are installed at once.

- **Disadvantages**

- A full package requires a larger amount of disk space on the workstation.
- Initial installation time is much longer compared to a partial workstation configuration.

Partial package

The partial package is a minimum OneWorld configuration. Using a partial package has the following advantages and disadvantages:

- **Advantages**

- A partial package requires much less disk space on the workstation compared to a full package.
- Initial installation times are much faster for a partial package. Only those objects required to begin using OneWorld are deployed during client

workstation installation. All other applications are delivered through Just-In-Time Installation when the user first enters an application.

- **Disadvantages**

There is a slight increase in network traffic when first-time users load applications. Performance for Just-In-Time Installation over a WAN may be slower than over a LAN. Developers cannot use partial packages since development objects are not delivered.

Update package

The update package complements a full or partial package enabling you to update, add, or refresh your package with modified objects. You can only deploy an update package to a workstation that already has OneWorld loaded. The objects in the update package replace those objects on the workstation. All other objects on the workstation are left untouched.

Unless the update package includes just-in-time applications, it is simply a point-in-time copy of your central objects for a particular path code. If the update package contains a just-in-time application, only that just-in-time application is dynamic. The rest of the package is static. In this case, dynamic implies the applications are pulled directly from the central objects location when the application is first selected at runtime.

The advantage of this type of package is that you can deploy software fixes or enhancements quickly.

Note

Developers must deploy full and update packages only since partial packages do not contain development objects.

13.1.1.2 Cumulative versus non-cumulative update packages

If you use a cumulative update package strategy, you have one package that you continually add to rebuild and to re-release to users, rather than creating a new package every time you have a modification you want to deploy. To use a cumulative package, you need to follow these steps:

1. Change the pre-existing update package status to add the new modifications.
2. Add the changed or new items to the package definition.
3. Rebuild the update package.
4. Redeploy the package.

If you use a non-cumulative strategy, you create and deploy a different package each time rather than using the same package. For example, if you deploy one modification per week for 10 weeks, there may be 10 different packages with only the software changes for that week.

13.1.1.3 Understanding Just-In-Time Installation

A partial workstation does not have any applications resident. Instead, applications are retrieved at runtime and loaded on the workstation the first time you select that application from the OneWorld Explorer menu. Loading happens only once. The next time you select the same application, it is already loaded on your machine contained in the OneWorld Table Access Management (TAM) specification files. This process is called “Just-In-Time Installation” (JITI).

Applications are loaded the moment you need them rather than requiring them to reside on your workstation. With the exception of Data Dictionary objects, JITI applies only to applications. Business functions cannot be installed through Just-In-Time Installation.

JITI also works when your workstation receives an update package containing a just-in-time application. In either case, when you make a menu selection from OneWorld Explorer, the runtime engine checks to see if the application Table Access Management (TAM) specifications reside on your workstation. In case they do not exist (and the environment Master Just-in-time installation flag is set to Y, and your security profile allows the application to be installed), the specifications are transferred from the Central Objects data source to the path code TAM files associated with your environment. This way, needed applications are installed “just-in-time” rather than when the user receives the package or at initial installation time. There is a slight delay during this JITI process. However, JITI occurs only the first time an application is selected that does not reside on the workstation. The next time you select that same application, there is no load delay because the application will reside on the workstation.

The update process works almost the same way. When you receive an update package containing a just-in-time application, OneWorld first checks to see if that application’s specifications reside on your workstation. If so, it deletes from your workstation old versions of that application. Then, the next time you select that application from the OneWorld Explorer menu, OneWorld loads the new version of that application. When a package includes just-in-time applications, at the time of execution, only the following related objects are deployed:

- Interactive or batch application specifications
- Embedded event rules for the application
- Processing option templates or data structures and related business views

The following objects are not deployed. Therefore, they must be included in the package if they were modified:

- Business functions and their data structures
- Generic text data structures
- Table event rules (comes with tables)
- Named event rules
- New icons

13.2 Recommendations and considerations

As a rule of thumb, you should build and deploy packages only when necessary. This alleviates the strain that is otherwise placed on the version control team. Such a measure helps to ensure a controlled planning and tracking process for the promotion of software modifications. We recommend that when you initially start this process, you should build and deploy packages on a set schedule.

Use the SAR system (or any numbering system) to track your modifications. Always *check in* these modified objects using an SAR number.

13.2.1 Working with path codes

Traditionally, the delivered OneWorld Object Configuration Master, environment settings, and path code settings are used in the manner described in the following list. You may adopt the same methodology or define your own.

- **DEVB733:** Use this path code for “normal” development. Upon successful testing, transfer the objects to your CRPB733 path code using Object Transfer. Distribute them to your users using the package build deployment process.
- **CRPB733:** This path code contains a practice set of objects that are tested during a Conference Room Pilot before transferring objects to production. It is for deploying quick fixes or making minor modifications that you quickly transfer to production. It can also be used as a place to test modifications that were done in the development path code before taking the risk of transferring them to the production path code.
- **PRODB733:** This is your production path code. Just-in-time installations come directly from this location. Production server objects are also deployed from here. After testing software changes in CRPB733, transfer them to PRODB733 and deploy the changes to your Enterprise Servers and workstations.
- **PRISTB733:** This is the set of pristine objects shipped from J.D. Edwards. You should not make changes to this path code other than paper fixes from J.D. Edwards. This path code is used to compare J.D. Edwards standard software to any custom solutions you have implemented in other path codes. You should keep a copy of this path code so that you have a “clean” copy of OneWorld in case you need to refresh anything.

All path codes share the same Object Librarian, system data source, and traditionally the same Data Dictionary tables. The only distinct tables across path codes are the central objects or specifications (F987*), the version list (F983051), the processing options text (F98306), and the user overrides (F98950).

13.2.1.1 Preserving the integrity of the production environment

As soon as you transfer objects into your PRODB733 path code, these changes can be accessed by end-users immediately. Therefore, you must make certain to test the modified objects before transferring them to production. There are two ways that objects transferred into the production path code can be immediately deployed to end-users:

- Anyone using *Just-In-Time Installation* (JITI) who uses an application for the first time has that application installed from the central object tables. This includes people who are using a partial package or anyone who has received an update package with a JITI item type.
- If you build an *update or partial package* and select the Build Business Functions check box for the indicated package, the business function included in that package is built and globally linked with all other business functions in the path code’s check-in location.

Avoid using partial packages or update packages with JITI item types when transferring modifications to your production path code. This prevents these changes from being immediately available to your end users. Also, do not transfer business functions into the production path code until you are ready to

deploy. Otherwise, a global build of business functions performed during a package build automatically includes the new functions.

By following these recommendations, you can be assured that when you transfer changes into the production path code, they will not be available until you build a full or update package.

13.2.1.2 Disabling JITI

Through Environment Master (P0094), you can disable Just-In-Time Installation (JITI) for a specified environment. Consequently, anyone who signs on to that environment and does not already have TAM specifications for an application cannot retrieve them because JITI is turned off. J.D. Edwards recommends that you use this flag during the cumulative installation process when you update your production central objects with the new J.D. Edwards changes. Any user who has installed a partial package or received an update package containing applications without specifications cannot access that application when JITI is disabled.

Even if you disable JITI, Data Dictionary items are still copied using just-in-time-installation to the global tables. This is due to the Data Dictionary structure and cannot be disabled.

13.2.2 Object promotion and version control

The typical application development process is shown in Figure 139 on page 332. It should help you understand the role of packages within this process.

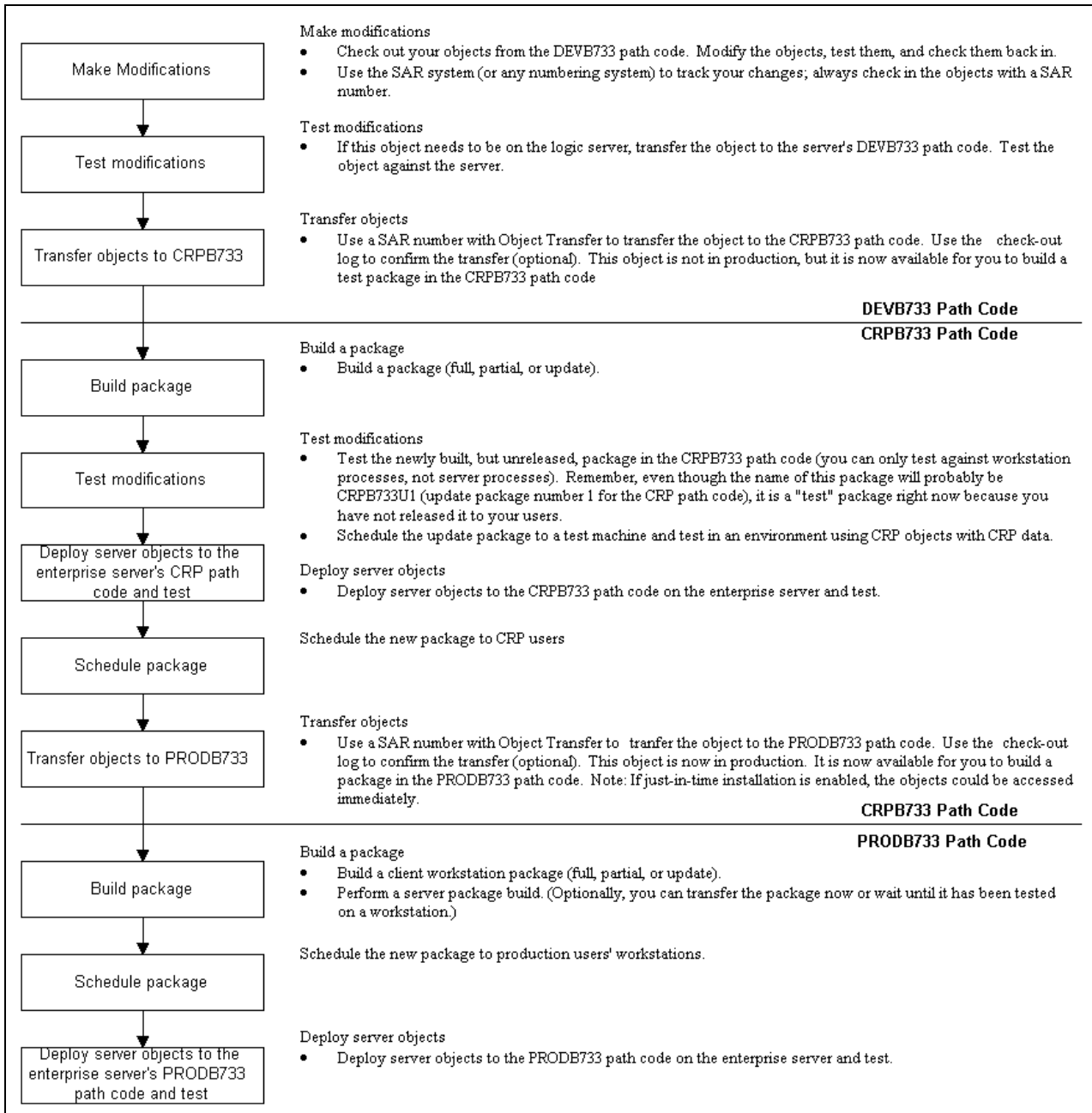


Figure 139. Typical development promotion schedule

13.3 Package deployment methodology

This section describes the package deployment methodologies. Each deployment method has its strengths and limitations. To help you decide which method is right for your needs, here are some key points about the different strategies:

- Workstation Installation is used to load a full or partial package to a new or existing OneWorld client workstation. You should also load any update packages required of users since the last full or partial package build.

- Package Deployment Instructions must be used to schedule the deployment of all update packages to workstations unless you are using the Push Installation feature.
- Package Deployment Instructions may also be used to schedule the deployment of full or partial packages if OneWorld has already been loaded on the workstation. You can use the Push Installation feature to deploy to a machine that does not have OneWorld installed.
- If workstation disk space is a factor, consider deploying a partial package that provides a minimum OneWorld configuration and allows users to load only the applicable OneWorld applications.
- Use Silent Installation to submit a Workstation Installation request through a command line argument. Do not use Silent Installation for an initial installation.
- Use Multi-tier Deployment to install from more than one deployment locations. You should consider this method if you have more than 50 workstations performing OneWorld software installations per day.
- Use Server Package Installation when you need to push objects in a server package from the Central Objects data source to an Enterprise Server.

13.3.1 Workstation Installation

The Workstation Installation program is used to deploy full and partial packages. Update packages cannot be deployed using Workstation Installation. Workstation Installation retrieves the items specified in the package. A full or partial package is like a bill of materials or a kit with instructions. The Workstation Installation program deploys to the workstation only the components that are specified in the indicated package.

Workstation installations are always initiated by the user. Users can deploy only those packages that are set up in their user or user group profile's deployment preferences. In a case where both the user and user group profile contain deployment preference instructions, the user's preferences are only available for selection.

Note: For workstations that previously installed OneWorld, it is possible to refresh the `jde.ini` file, workstation registry, OneWorld security files, and ODBC's. To do so, run the Workstation Installation program in a "NOCOPY" mode. Click the Windows **Start** button, and select **Run**. Then, type the following command:

```
\\<deployment_server_name>\b733\client\setup.exe -n
```

When launching the Workstation Installation program using this syntax, the typical displays and dialog boxes appear. Select a package that was previously deployed to your workstation, for example, CPRB733FA. A dialog box appears that asks the question: `Directory x:\b7\CRPB733 exist, would you like to erase contents?` Click **No**. The installation completes shortly thereafter and refreshes your `jde.ini` file, workstation's registry, OneWorld security files, and ODBC's (as defined in the `x:\oneworld\b733\client\ODBCDataSource.inf` Deployment Server file).

13.3.2 Package Deployment Instructions

Package Deployment Instructions enable the OneWorld administrator to specify the date and time when a package is made available to the affected users or user

groups. The administrator can specify whether the package is mandatory or optional. If a package is mandatory, users who receive the package cannot access OneWorld until they load the mandatory package.

Users who receive a scheduled package are given the opportunity to load the package immediately after they sign on to OneWorld on the specified date. If they choose to load the package, the installation routine launches, and the package loads. The user can also choose to load the package later or decline installation altogether (unless the package is mandatory).

As an alternative, the push installation feature enables the administrator to schedule a package that is automatically “pushed” from the Deployment Server to the workstations at the scheduled time, without requiring any interaction with the user.

Update packages typically use Package Deployment Instructions. The full and partial package types are normally deployed using Workstation Installation. However, you can use package deployment for a full or partial package in cases where OneWorld is already loaded on a machine and you want to redeploy. See “Package Deployment Instructions” in the *OneWorld Package Management Guide* for more details.

13.3.3 Multi-Tier Deployment

Multi-Tier Deployment enables OneWorld workstations to install software from more than one deployment location and more than one deployment machine. J.D. Edwards recommends that you consider Multi-Tier Deployment if your site has more than 50 workstations performing OneWorld software installations per day, or if you are deploying OneWorld software across a WAN connection. See “Multi-Tier Deployment” in the *OneWorld Package Management Guide* for more information.

13.3.4 Recommendations for sites using partial packages and JITI

J.D. Edwards recommends that you adopt a non-cumulative update package strategy. Each week that you build a new update package, if you have a modification to deploy that cannot be done using Just-In-Time Installation (such as business function changes), we recommend that you also rebuild the partial package for that week to ensure that the partial package contains the latest changes. If you follow this recommendation, when you want to load a new workstation (or completely refresh any machine), you simply need to install the partial package. You do not need to install any update packages.

13.3.5 Recommendations for sites using full packages and JITI

J.D. Edwards recommends that you adopt a cumulative update package strategy. Each week that you need to deploy a change, add that object to the existing update package, and then rebuild and schedule it. The advantage of this strategy is that you do not need to rebuild your full package each week. By using this strategy to load a new workstation (or to completely refresh any machine), you need to install the full package and the one cumulative update package.

The disadvantage of using this strategy is that the update package may become so large that deployment time is increased. You need to determine your threshold

for when to rebuild the full package that the new workstations will load, while existing users install the new update package.

13.3.6 Related topics not covered in this chapter

For detailed information pertaining to the following topics, refer to the *OneWorld Package Management Guide*. The topics not included in this chapter are:

- **Transferring objects:** The Object Transfer application enables OneWorld administrators to transfer objects just as the Object Librarian does. However, Object Transfer transfers objects from the central objects for one path code to the central objects for another path code. The Object Librarian transfers objects from the workstation to the central objects location, and vice versa.

See “Transferring Objects” in the *OneWorld Package Management Guide* for more details.

- **Checkout Log:** The Object Librarian logs all checkin, checkout, and delete operations to a Checkout Log. This log is stored in the system data source in the Checkout Log table - F9882. The Work with Checkout Log Application (P9882) allows developers and version control personnel to view the Checkout Log.

See “Checkout Log” in the *OneWorld Package Management Guide* for more information.

- **Cross Reference Facility:** You can use the Cross Reference Facility to obtain information about how and where specific types of objects are used. You can also view relationships between those objects and their components.

See “Cross Reference Facility” in the *OneWorld Package Management Guide* for more details.

- **Managing objects with the SAR System:** You can manage change control any way you want. At J.D. Edwards, the Software Action Request (SAR) system is used. Currently, the SAR system exists only in WorldSoftware. In the future, it will be available in OneWorld. Any numbering system will work for change control (you can even use a paper system with assigned numbers if you want).

See “Managing Objects with the SAR System” in the *OneWorld Package Management Guide* for more information.

13.4 Custom modifications

It is important that you understand which types of objects (and, therefore, modifications) can be deployed through each package type. Partial packages include:

- All consolidated business function dynamic linked libraries (DLLs)
- Specifications for only those applications necessary to sign on and get to OneWorld Explorer
- All icons
- Help, data, and foundation

Table 21 illustrates which types of changes are installed with a full package, an update package with Object item types, and an update package with JITI item types.

Table 21. Modifications installed by package types

Modification	Full package with object type (no JITI items)	Update package with JITI type	Update package
Applications:			
Imbedded Event Rules	X	X	X
Vocab Overrides (FDA text)	X	X	X
Data Structure	X	X	X
Process Options (Report)	X	X	X
Business Functions:			
C Language Source/Include/Object (if there is a compiler)	X	X	
Consolidated BSFN DLLs	X	X	
Data Structure	X	X	
Table ER	X	X	
Named ER	X	X	
Batch Applications:			
Report	X	X	X
Imbedded ER in Report	X	X	X
Report Data Structure	X	X	X
Report Vocab Overrides	X	X	X
Report Processing Options	X	X	X
Versions and Processing Option Values (depends on processing options)	X	X	X
Imbedded ER in Versions	X	X	X
Processing Option Templates	X	X	X
Business Views:			
Fields Added/Changed	X	X	X
Tables:	X	X	
Structure (specifications)	X	X	
Indexes	X	X	
Joins	X	X	
Miscellaneous:			
Helps Stored on the Server			

Modification	Full package with object type (no JITI items)	Update package with JITI type	Update package
Helps Stored on the Workstation	X	X	
Generic Text Data Structure	X	X	
Data Dictionary Items (see Data Dictionary Administration in the <i>System Administration</i> Guide)	X	X	X
Foundation Code (foundation is required for full and partial packages, and option for update packages)	X	X	X
Foreign Languages	X	X	X
Non OneWorld Objects (custom items can be deployed through any package type)	X	X	X
Replicated Local Data (data is required for full and partial packages and optional for update packages)	X	X	X
New Icons	X	X	

13.5 OneWorld modification rules

Because the OneWorld Tools are comprehensive and flexible, you can customize certain aspects of business solutions and applications without making custom modifications. J.D. Edwards refers to this concept as “modless mods”. These are modifications that you can perform easily without the help of a developer. You can perform modless mods on:

- User overrides
- User-defined codes
- Menu revisions
- All text
- Processing option values
- Data Dictionary attributes
- Workflow processes

This flexibility improves efficiency and provides distinct advantages, such as the ability to:

- Export grid records to other applications, such as Microsoft Excel spreadsheet
- Re-sequence a grid on a different column
- Change grid fonts and colors
- Control major functionality using processing options

When modifying OneWorld, you should adhere to certain rules and standards to ensure a smooth upgrade to the next release level. You should prepare for the upgrade before making any custom modifications.

OneWorld keeps track of all custom modifications as you check them into the server. Using this feature, you can run the Object Librarian Modifications (R9840D) report prior to a merge to identify the changed objects.

OneWorld consists of control tables (such as menus, user-defined codes, versions, and the Data Dictionary) and transaction tables (such as Address Book and the Sales Order File). J.D. Edwards ships the control tables with data that you can modify. The transaction files contain your business data. Both of these kinds of tables go through an automatic merge process during an upgrade where control tables are merged with new J.D. Edwards data. Transaction files are converted to the new specifications, preserving your existing data. For the object specification merges (such as business view, tables, data structures, processing options, event rules, and applications), there are processes in place that merge the specifications or overlay them depending upon the rules defined under “What an Upgrade Preserves and Replaces”.

What is preserved during an upgrade?

If your business requires custom modifications, observe the following general rules for OneWorld modifications to help ensure a smooth and predictable upgrade. These rules describe which of your modifications the upgrade process preserves and which modifications it replaces:

- *Preserve* means that during an upgrade, OneWorld automatically merges your modifications with the new J.D. Edwards applications shipped with the upgrade. You do not lose your modifications. If there is a direct conflict between your specifications and J.D. Edwards specifications, the upgrade process uses yours. When there is no direct conflict between the two, the upgrade process merges the two specifications.
- *Replace* means the upgrade does not merge those types of modifications. Therefore, J.D. Edwards replaces your modifications. You need to perform them again after the upgrade completes.

13.5.1 General rules for modification

The following general modification rules apply to all OneWorld objects:

- When adding new objects, use system codes 55 to 59. OneWorld uses reserved system codes that enable it to categorize different applications and vertical groups. By using system codes 55 to 59 for your custom usage, OneWorld does not overlay your modifications with J.D. Edwards applications.
- Do not create custom or new version names that begin with ZJDE or XJDE. These are reserved prefixes for standard version templates that J.D. Edwards delivers, these may be copied to create your own templates or versions. Using these prefixes does not preserve your custom versions in the case of a naming conflict.
- For upgrades, you should build a package from the last modified central objects set. Then, perform backups of your Deployment Server, central objects, and Object Librarian data sources so you can access those specifications for comparison or for troubleshooting purposes. See the *OneWorld Upgrade Guide* for information.
- For details regarding modification rules, what is preserved, and what is replaced for each of the following topics, see the *OneWorld Package Management Guide*:
 - Interactive Application Rules
 - Report Rules
 - Application Text Changes
 - Table Specification Rules

- Control Table Rules
- Business Views Rules
- Rules for Event Rules
- Data Structure Rules
- Business Functions Rules
- Version Rules

13.5.2 Checking the environment before building packages

Before you start building packages, check the environment by performing following steps:

1. Ensure that all server names can be found under the MMMKey in the F9650 table. This table is the Machine Detail Master and contains all machines with OneWorld installed that have connected at least once to the system database and successfully registered themselves or machines that have been added to the system using P9650a or the planner. It is not necessary to verify that all workstations are in this table. This table can be found in the system data source and the default owner is SYSB7xx, where xx is the base release of OneWorld.
2. Validate all network connections by issuing the ping command at the command line. On Windows platforms, click **Start->Run**. Then type:

```
ping <machine name>
```

At the command line on the AS/400 system and UNIX, type:

```
ping <machine name>
```

Note

If you ping the server from itself, you will return the IP address for the primary network interface card (NIC). This should match the IP address returned from a workstation ping of the same server.

3. Verify the number of header and source files (c components) on the Deployment Server, in the check-in location: \\<deployment server>\<release (for example, B733)>\<pathcode>\include or source. Source objects will have the extension *.c (source files), and include objects will have the extension *.h (header files). OneWorld Release B733.2 should have at the minimum 4,713 source objects and 6,677 include objects present.
4. Using a network sign-on that you will use when building packages, verify that you can add a directory to the Deployment Server under the Package directory (\\<deployment server>\<release>\<pathcode>\Package). In addition, make sure you can copy objects from the check-in location (see step 2) to this new directory and back. Lastly, make sure you can create a directory or library on your logic or Enterprise Server and that you can copy objects there from your package directory on the Deployment Server.
5. Check to see that you have Microsoft Visual C++ loaded on the Deployment Server or workstation from which you will build packages. You also must have ANSI C compiler loaded on any logic server (including the Enterprise Server). For a Windows NT logic or Enterprise Server, make sure that the following environment variables exist for "System" (not just "User"). These are found by going to Control panel, and double-clicking the **System** applet. Then, click the

Environment tab. Either add the following in or change the existing entries to match. The server must be re-booted if you change any paths.

Note: It is possible to have these set under “User” only, depending on how the customer is signing on the machine. Having them under “System” should always work. Also, these paths assume C++ has been installed into the default directories.

```
INCLUDE:  
c:\msdev\vc98\alt\include;  
c:\msdev\vc98\mfc\include;  
c:\msdev\vc98\include;  
LIB:  
c:\msdev\vc98\mfc\lib;  
c:\msdev\vc98\lib;  
PATH:  
c:\msdev\Common\Tools\WinNT;  
c:\msdev\Common\MSDev98\Bin;  
c:\msdev\Common\Tools;  
c:\msdev\vc98\bin
```

6. Verify the jde.ini on the logic or Enterprise Server (for a server package) or the workstation or Deployment Server (wherever you are building a client package) has the correct path to the ANSI C Compiler. This is in the jde.ini [JDE_CG] section.

Note: No spaces are allowed in the path.

7. Verify that the jde.ini [BSFN Build] BuildArea path on a logic or Enterprise Server is correct. Access the server and make sure the path to which the BuildArea refers exists on the server.
8. Make sure R9621 and R9622 are mapped through OCM to the LOCAL logical data source.
9. Verify that porttest.exe works (Porttest is documented in the OneWorld Server and Workstation Administration Guide). It is a system executable shipped with OneWorld, and requires a system user, system password, and environment name to run.
10. Verify that you can run a UBE on the server (example R0006P).
11. If you are going to use a security server, try using porttest and running a UBE with security on.
12. Verify the record counts for Central Objects. Table 22 can help you to verify your record counts for the Central Objects table and their associated Spec file.

Note

If a customer has done any custom coding, their counts may exceed these numbers, but should never be less.

Table 22. Record count information

Table name	Table description	SPEC name	Record count
F98710	Table Header	DDTABL	1,879
F98711	Table Columns	DDCLMN	58,264
F98712	Primary Index Header	DDPEKEYH	3,849
F98713	Primary Index Detail	DDPKEYD	14,319
F98720	Business View Specifications	BOBSPEC	3,215
F98740	Event Rules – Link Table	GBRLINK	56,664
F98741	Event Rules – Specification Table	GBRSPEC	1,610,571
F98743	Data Structure Templates	DSTMPL	12,895
F98745	Smart Field Named Mappings	SMRTTMPL	55
F98750	Forms Design Aid Text Information	FDATEXT	94,336
F98751	Forms Design Aid Spec Information	FDASPEC	189,909
F98752	Forms Design Aid/Software Versions	ASVRHDR	1,694
F98753	Forms Design Aid/Software Versions	ASVRDTL	4,313
F98760	Report Design Aid Text Information	RDATEXT	115,449
F98761	Report Design Aid Specification Info	RDASPEC	191,925
F98762	JDEBLC – Behavior Information	JDEBLC	6,294

13.6 Defining and building a package

Packages allow you to deploy software changes and new applications to your users or to install OneWorld on a workstation for the first time. After you define and build a package, you can deploy it using the Client Workstation Installation application or Package Deployment Instructions.

Note

J.D. Edwards delivers standard full and partial packages for your use. If you do not have extensive modifications, consider using these packages. Update packages containing your modifications may then be built to supplement these.

13.6.1 Working with package names

J.D. Edwards recommends defining an A and B version of a package. This allows you to alternate between the two, making one available for deployment while the other is being built. Then, each time you build a package, you alternate which package you build.

Using this scenario if you are using both full and partial packages, you have four packages per path code. This setup gives you two full packages (A and B) and two partial packages (A and B) for production, for example:

- PRODB733FA Standard Production Full A
- PRODB733FB Standard Production Full B
- PRODB733PA Standard Production Partial A
- PRODB733PB Standard Production Partial B

Cumulative Update packages may be named:

- PRODB733UA Production Update Package 1
- PRODB733UB Production Update Package 2
- PRODB733UA Production Update Package 3
- PRODB733UB Production Update Package 4

Non-cumulative Update packages may be named:

- PRODB733U1 Production Update Package 1
- PRODB733U2 Production Update Package 2
- PRODB733U3 Production Update Package 3
- PRODB733U4 Production Update Package 4

13.6.2 Overview of creating and deploying a package

The following overview presents the steps involved in creating and deploying a package:

1. **Assemble the package:** The initial step involves specifying the type of package you are building and providing a name, path code, and package description. Next, you assemble your package by specifying the objects you want to include in the package. If you are building a partial or update package, you can specify individual objects to include.

The process of assembling the package is simplified by the package Assembly Director, which displays a series of forms that guide you through the steps of naming your package and assembling the objects you want to include in the package.

2. **Define the package build:** After you assemble the package, you must define the build before you can deploy the package to your workstation and servers. In this step, you specify build options, business functions build options, and compression options. You also need to specify whether the package is for a

workstation, server, or both. If the package is for servers, you must specify which servers should receive the package.

To simplify the build process, the Package Build Director displays a series of forms that guide you through the steps of specifying where to build the package, whether to include specifications, whether to compress or build business functions, and so on.

3. **Build the package:** During the actual build process, OneWorld takes the information you provided when you assembled and defined the package and copies and converts central objects to the package. It also performs a global build of the business functions included in the package and then compresses the package.
4. **Schedule the package for deployment to workstations:** If you build an update package (or if you want to redeploy a full or partial package to a workstation that already has OneWorld), you must specify the date and time to deploy the package. When you schedule the package, you can make package installation mandatory or optional.
5. **Deploy the package to Enterprise Servers:** Use Package Deployment to move any changed objects to the Enterprise Server.

If you specify a server during the package build definition process, OneWorld automatically creates a corresponding server package in the correct format. If you do not specify a server and define only a workstation package, you should create a corresponding server package. The process is nearly identical to creating a workstation package.

Note: J.D. Edwards recommends that you build a client and server at the same time when building server packages.

13.6.3 Understanding the Package Assembly Director

The package assembly process is greatly simplified by the Package Assembly Director, which displays a series of forms that step you through the process. During Package assembly, the status will always be either In Definition or Definition Complete.

The Package Assembly Director's forms guide you through the process of specifying or confirming the location where package components can be found, as well as indicating the objects to include in the package. Throughout the package assembly process, you always have the option of going to the previous or next screen by clicking Previous or Next. Also, regardless of where you are in the process, you can always cancel the assembly process by clicking Cancel.

13.6.4 Accepting default values

Many of the Package Director's forms have a default value and, after making sure you want to use the displayed value, you can advance to the next form without entering anything.

Forms determine the default values based on:

- **Foundation:** The default foundation location is the server share path under the path code associated with the package.
- **Help:** The default help location is the server share path under the path code associated with the package.

- **Database:** The default database location is the server share path under the path code associated with the package.
- **Objects:** The default location for full packages is the deployment data source.
- **Language:** The default language is English.

On forms that have a default value, even if you change or clear the field, you can always restore the original default value by clicking the Default button.

If you are building a full or partial package and do not need to specify the objects in that package, the fastest way to define the package is to accept the default locations for foundation, database, help, and language. This method applies only to full and partial packages. For an update package, if you accept the defaults but do not include any objects, you will have an empty package.

13.6.5 Assembling a package

The following task describes how to assemble a package using the Package Assembly Director:

1. From the Package and Deployment Tools menu (GH9083), choose **Package Assembly** (P9601). The Work with Package form appears (Figure 140).

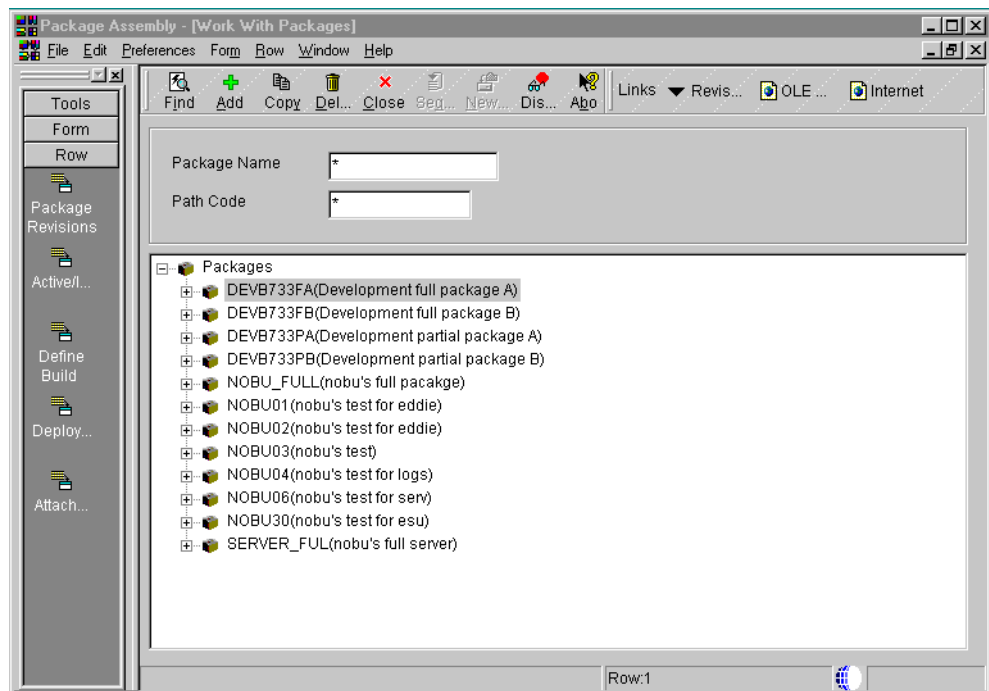


Figure 140. Work with Packages form

Any previously assembled packages will be displayed on this form. As on any other parent or child form, you can expand or compress a package listed by clicking the plus (+) or minus (-) symbols to show more or less information.

For any previously assembled packages, underneath the package name you can view the package properties (including package type and current status), as well as the selections for foundation, database, help, and language.

2. To assemble a new package, click **Add**. The Package Assembly Director launches (Figure 141).

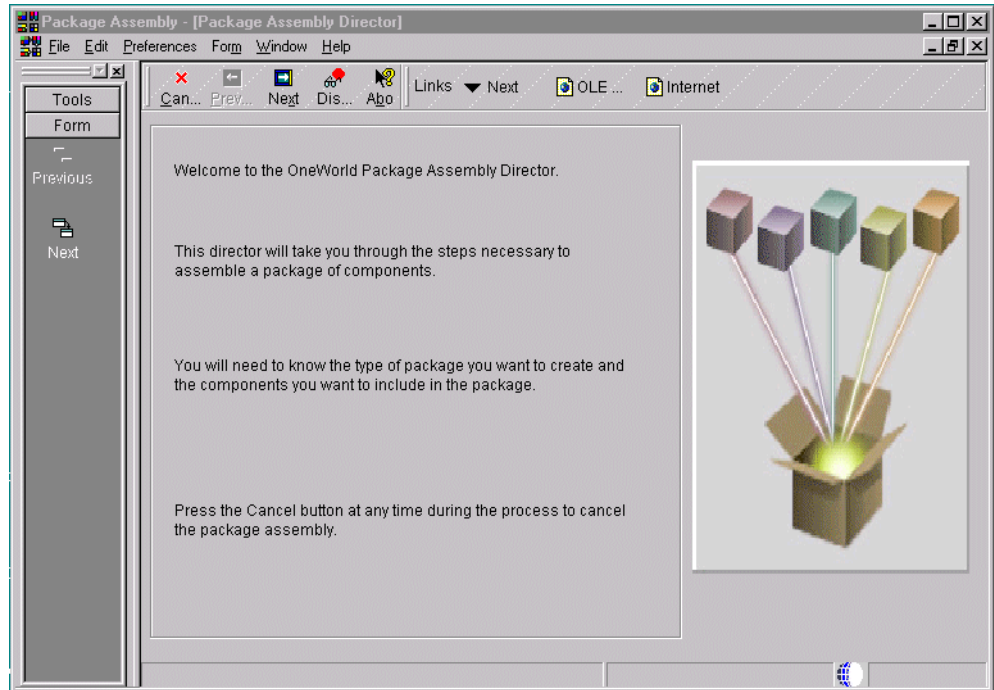


Figure 141. Package Assembly Director

3. On the Welcome display, click **Next**. The Package Information form appears (Figure 142).

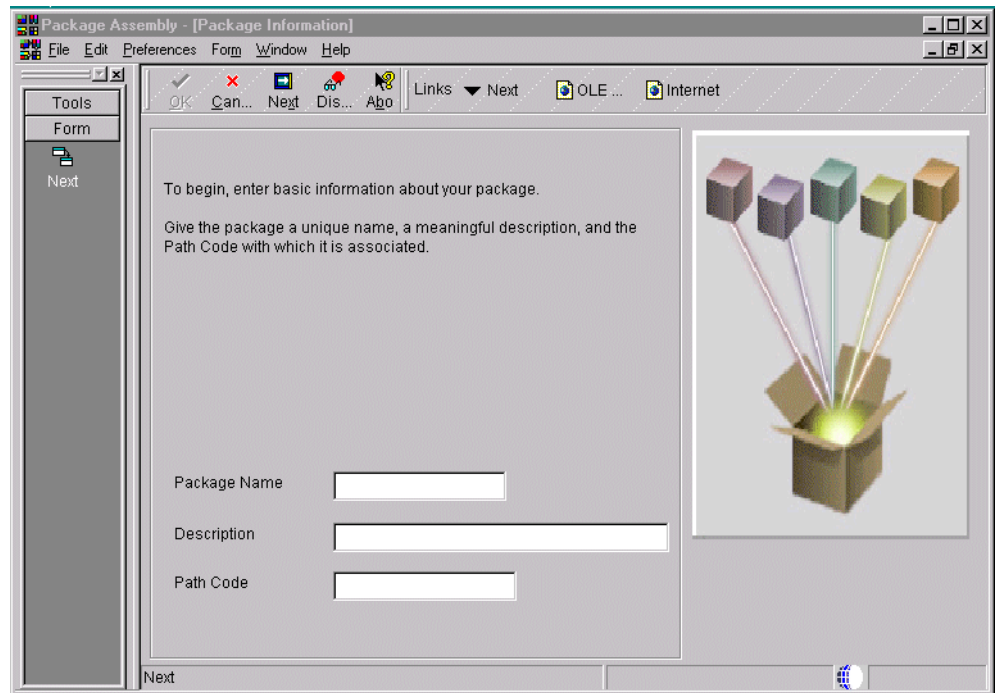


Figure 142. Package Information form

Complete the following fields:

- Package Name
- Description
- Path Code

4. Click **Next**. The Package Type Selection form appears (Figure 143).

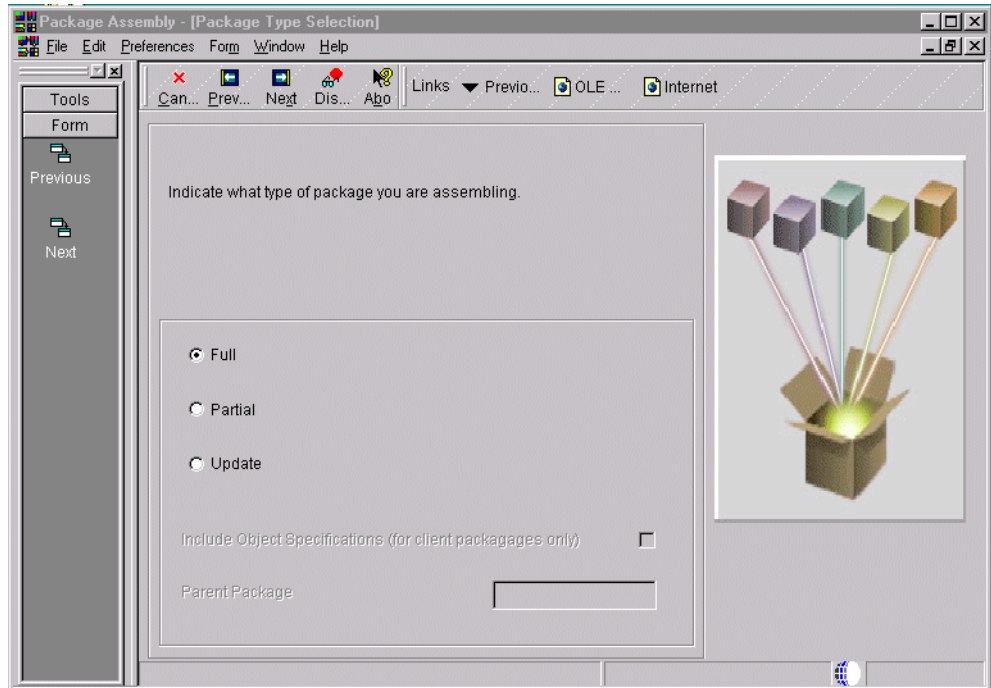


Figure 143. Package Type Selection form

Complete the following fields:

- Full, Partial, or Update
- Parent Package (update packages only)
- Include Object Specifications (update packages only)

Be sure to click **Include Object Specification** whenever you assemble an update package that will go to one or more servers.

5. Click **Next**. The Foundation Component form appears (Figure 144).

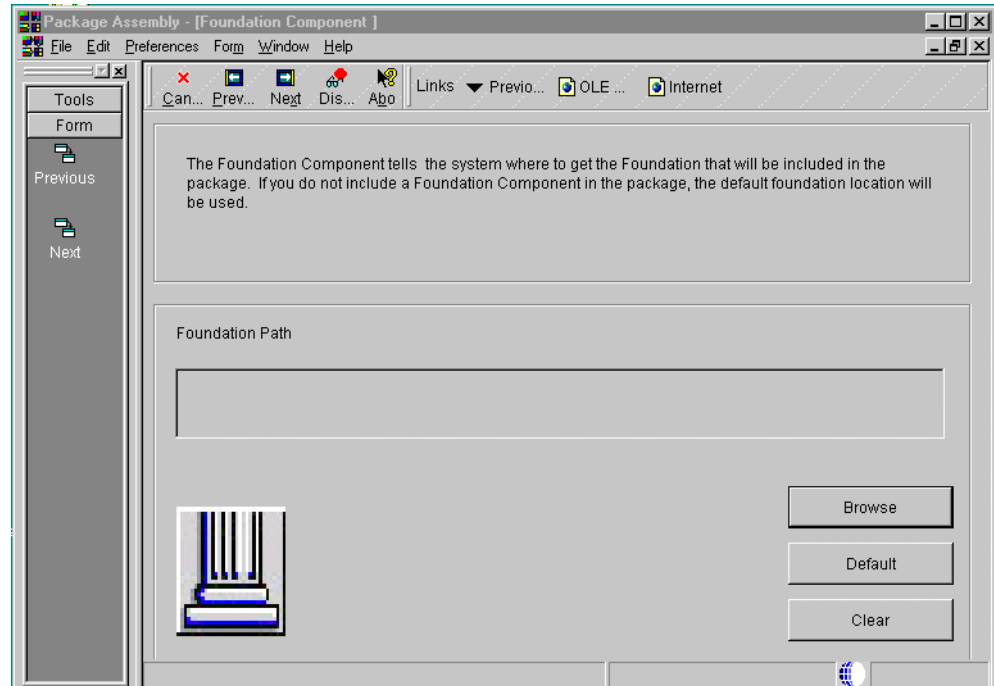


Figure 144. Foundation Component form

Accept the displayed location by clicking **Next**, or click **Browse** to specify another foundation location. To clear the currently displayed location, if any, click **Clear**. The Clear option applies only to update packages.

6. Click **Next**. The Help Component form appears (Figure 145).

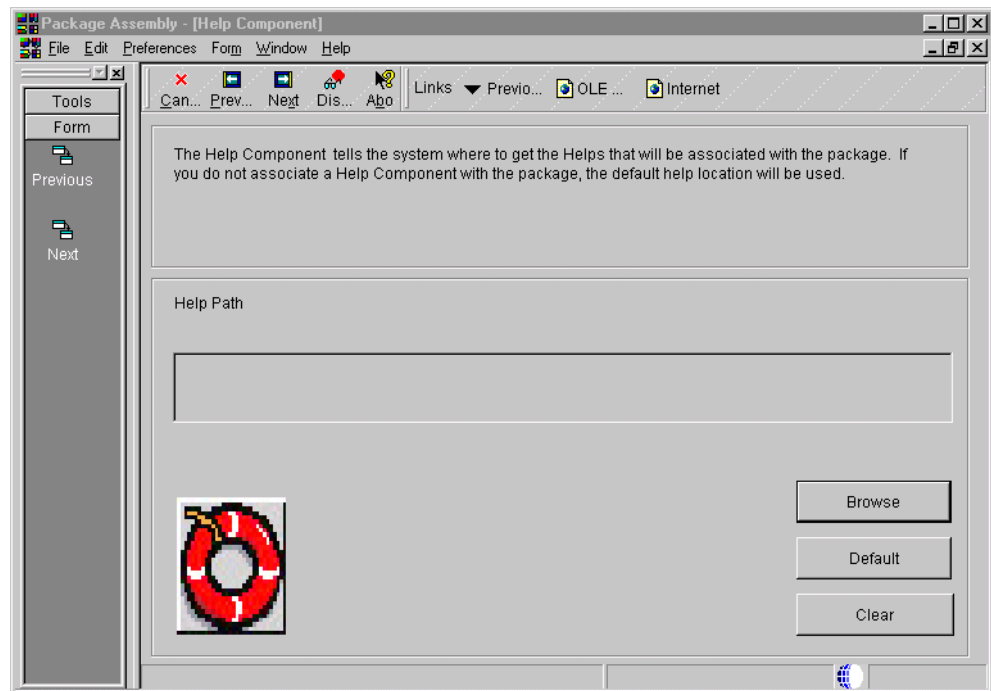


Figure 145. Help Component form

Accept the displayed default location by clicking **Next**, or click **Browse** to specify another help location. To clear the currently displayed location, if any, click **Clear**. The Clear option applies only to update packages.

7. Click **Next**. The Database Component form appears (Figure 146).

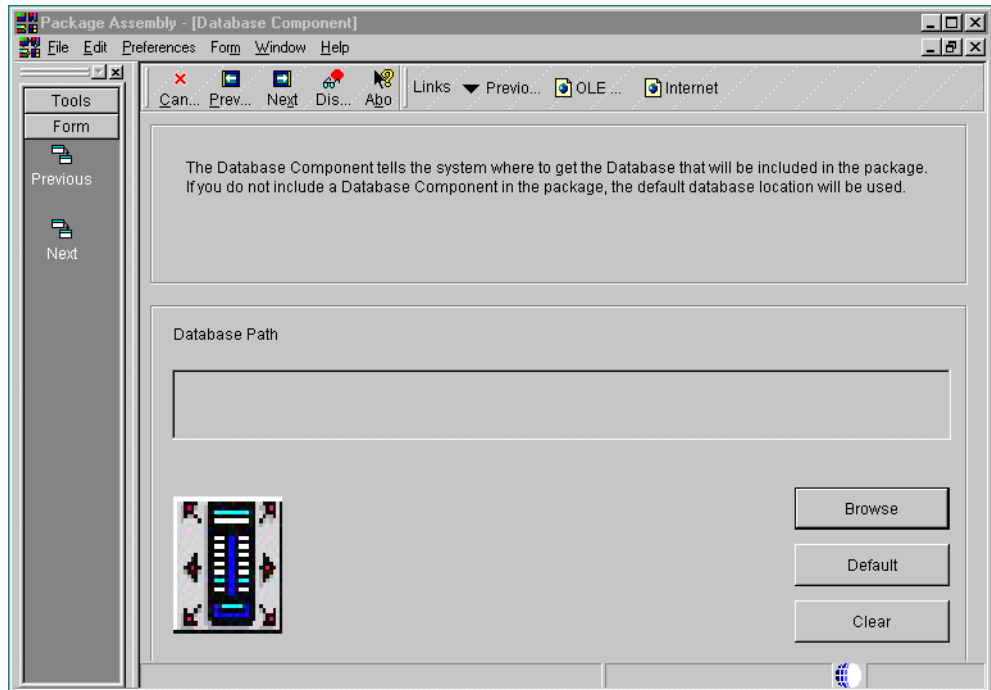


Figure 146. Database Component form

Accept the displayed default location by clicking **Next**, or click **Browse** to specify another database location. To clear the currently displayed location if any, click **Clear**. The Clear option applies only to update packages.

8. If you are assembling a full package, click **Next**. The Default Object Component form appears. If you are assembling a partial or update package, skip this step and proceed to the next step.

OneWorld will build your package from the deployment data source associated with the displayed default object path. Verify that the correct location is displayed. Because you are assembling a full package, skip the following step.

9. Click **Next**. When you are assembling a partial or update package, the Object Component form appears (Figure 147). This form allows you to specify the individual objects you want to include in your package. When you revise a previously-assembled package, objects you added earlier are displayed.

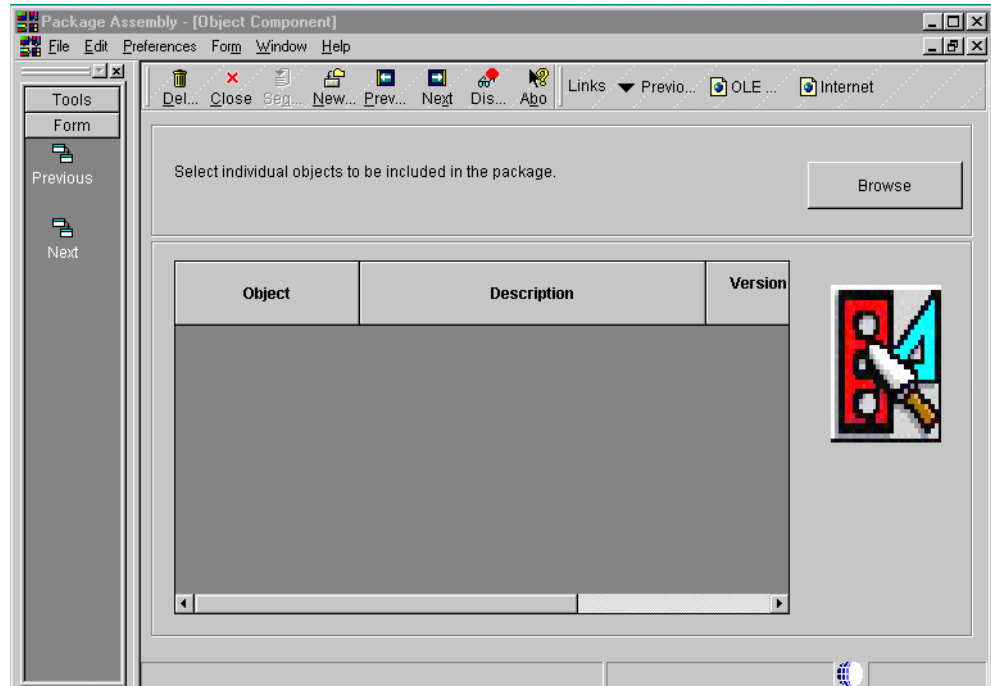


Figure 147. Object Component form

To add an object, click **Browse** to display the Object Component Selection form. Use this form to locate and select the objects you want to include in your package. When you are finished adding objects, click **Close** to return to the Object Component form.

For more information about adding objects, refer to 13.6.6, “Adding objects to a package” on page 351.

10. Click **Next** to display the Language Component form (Figure 148 on page 350). This form allows you to add to your package language specifications for a language other than English. If you add a language to your package, only that language will be included. For example, if you add French, English will not be included.

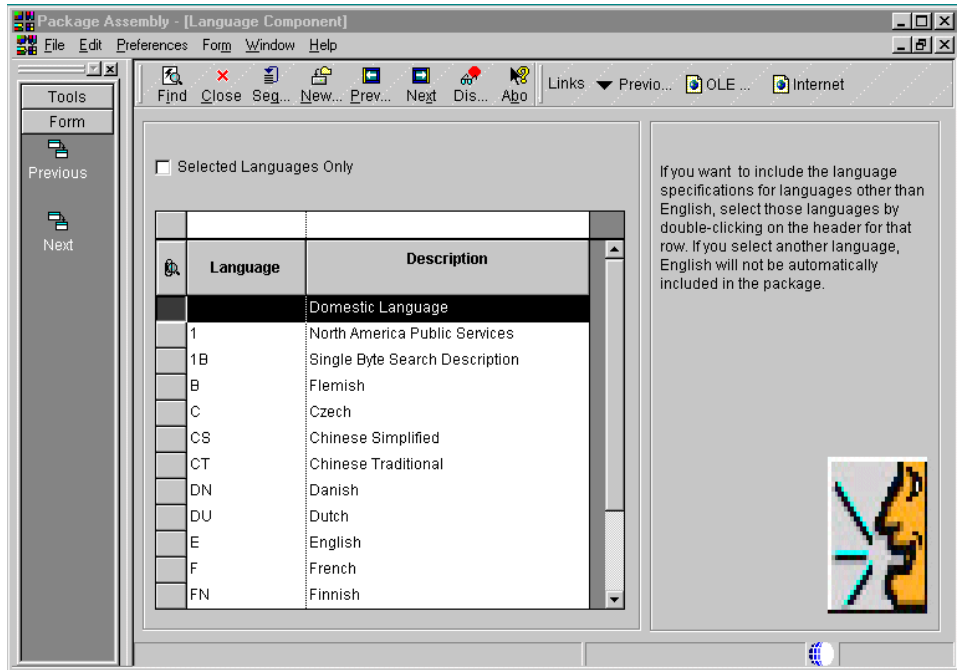


Figure 148. Language Component form

11. Click **Next** to display the Package Component Revisions form (Figure 149). This form allows you to see at a glance the current foundation, help, and database locations, as well as the objects in the package and any language selection.

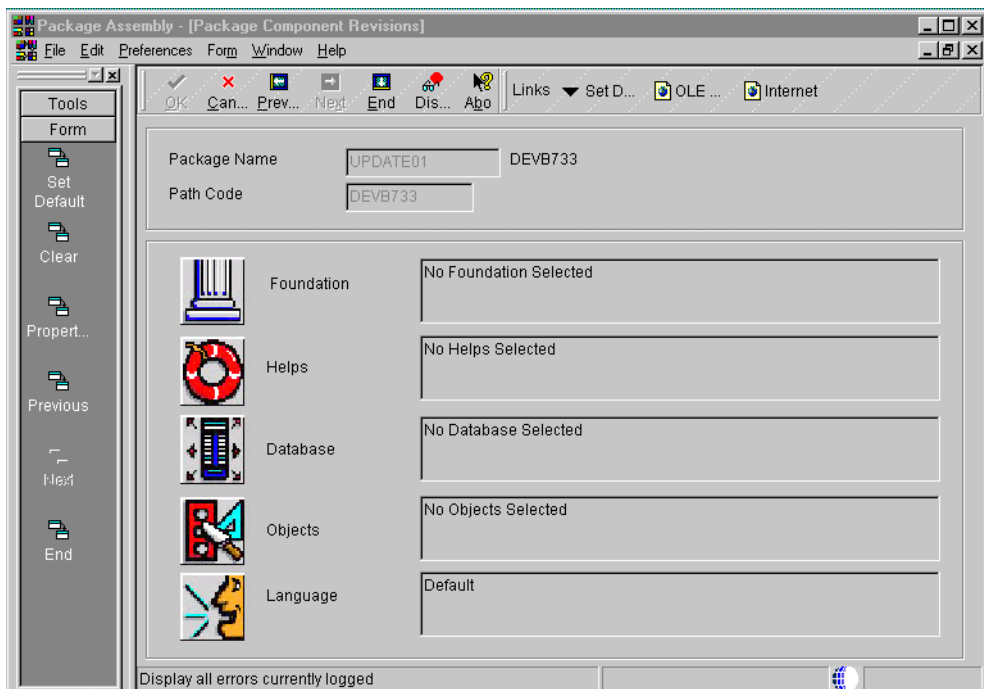


Figure 149. Package Component Revisions form

To change any of these package components, click the icon for the component you want to change. The form for that package component appears.

12. When you are finished assembling the package, click **End** to exit from the Package Assembly Director. At this point, you are ready to change the package's status to [assembly] Definition Complete as described in 13.6.7, "Activating a package" on page 352.

13.6.6 Adding objects to a package

If you are assembling an update or partial package, you can select individual objects to include in the package. When you are finished adding objects, those objects will be displayed on the Package Assembly Director's Object Component form, the Package Component Revision form, and the Work with Packages form.

The following task describes how to add objects to a partial or update package. The procedure is the same whether you are adding objects for the first time or revising a previously assembled package.

1. Display the Object Component [Section] form (Figure 150) by doing one of the following tasks:
 - On the Object Component form, click **Browse**.
 - On the Package Component Revisions form, click the **Objects** icon to display the Object Component form, and then click **Browse**.

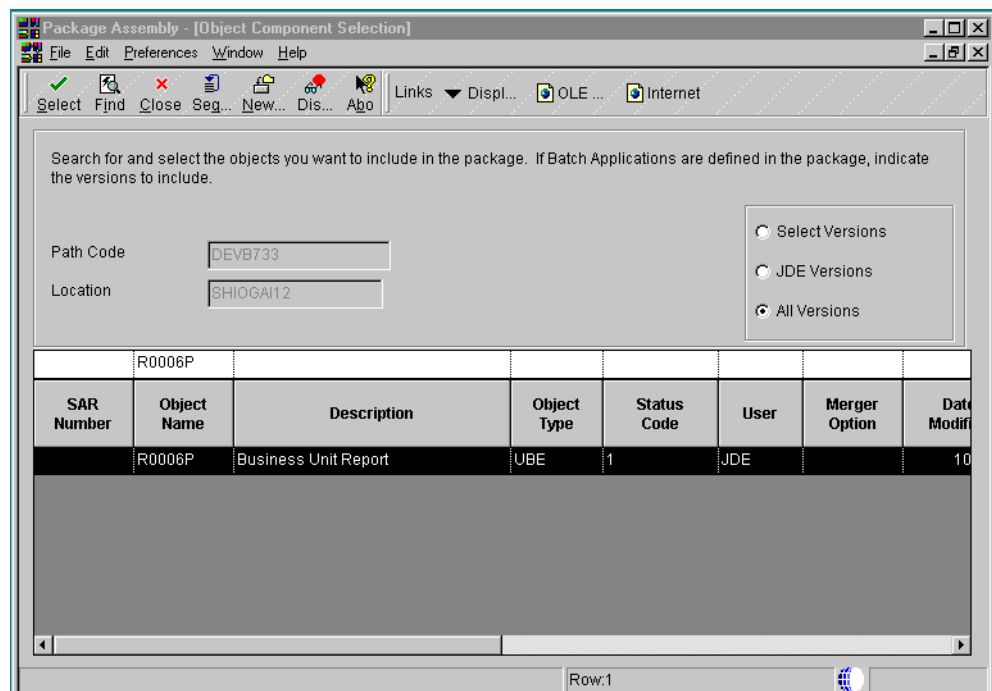


Figure 150. Object Component form

2. Find and choose the object you want to include in the package, and then click **Select**.
3. If you are adding a batch application, the Version Selection form appears so you can specify the appropriate version.

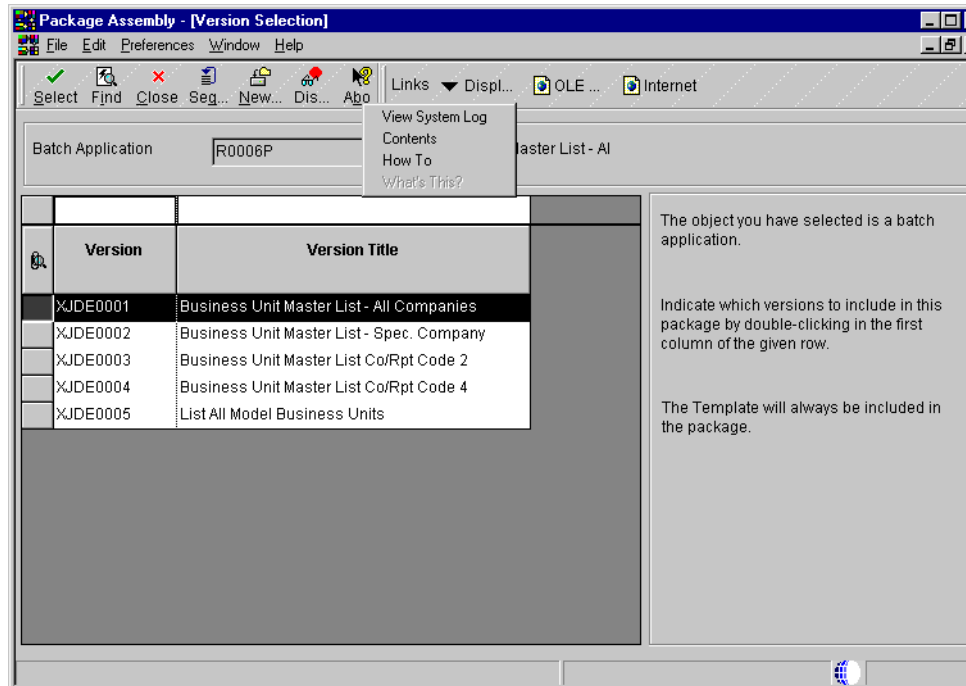


Figure 151. Version Selection form

According to your preference, click **All Versions** or **JDE Versions**, or choose the specific individual versions you want to include. The list of versions appears. Note that only versions that have been checked in are displayed.

When you are finished, click **Close** to return to the Object Component Selection form.

4. Repeat this process until you have finished adding objects to your package. When you are finished, click **Close** to return to the place from which you accessed the Object Component Selection form.

13.6.7 Activating a package

After you have you have assembled a package, the status remains [assembly] In Definition, which means the package is inactive, until you specifically change it to [assembly] Definition Complete, which means the package is active. An assembled package cannot be built until the status has been changed to Definition Complete. Changing a package's status to Complete indicates to the system that you are finished assembling the package and are ready to begin the build definition process.

To activate a package, perform following steps:

1. From the Package and Deployment Tools menu (GH9083), choose **Package Assembly** (P9601). The Work with Packages form appears.
2. From the Work with Packages form, choose the package you want to activate. Packages that are currently In Definition are identified by an open box icon, while packages that have a Complete status have a closed box icon.
3. Choose **Active/Inactive** from the Row menu.

You can use this same process to change a Complete package back to In Definition. Now you are ready to Define the Package Build.

Refer to the *OneWorld Package Management* manual for information on:

- Revising an existing package
- Copying a package
- Deleting a package

13.6.8 Defining a package build

After you assemble a package, you must define the package build before you can deploy it to your workstations. The build process reads the Central Objects data source for the path code you defined in the package. This information is then converted from a relational format to replicated objects, which are put in the package itself.

On the Deployment Server, the B733 (release name) directory structure looks similar to the display in Figure 152.

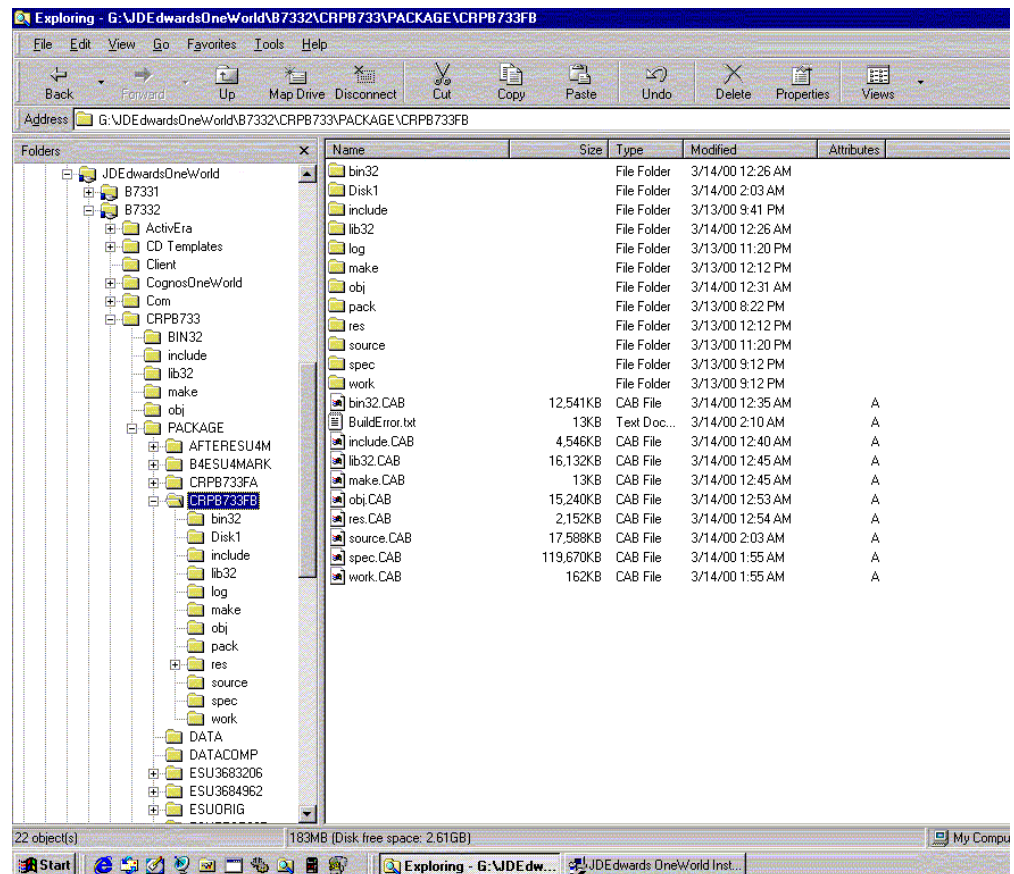


Figure 152. OneWorld Deployment Server package name directory structure

When you build a package, the directories under the package name are populated. Information for the source and include directories is copied from the location under the path code on the Deployment Server from which developers check in development objects. Information for all other directories comes from the Central Objects data source. The bin32, lib32, and obj directories are populated with the output of the business function build process.

13.6.9 Building business functions during a package build

As part of the package build process, you can specify whether you want to build business functions. If so, after the package is built, a global build of business functions takes place.

When building full or partial packages, you should always build business functions. If an update package includes a business function object, you should build business functions.

The process of building business functions as part of the package build is the same as if you manually execute the BusBuild program after the package is built, and select Build from the Global Build menu. During the BusBuild process, rebuilding fine tunes the performance of DLLs so that they run faster.

Source and header information is gathered from the package (from the source and include directories), compiled, and put into the bin32, obj, and lib32 directories. Business functions are built in the package, not on the workstation. After business functions are built, they are compressed if you turned on the Compress Package option.

Enter the information needed for the business function build through processing options. For more information, see “Package Revisions Processing Options” in the *OneWorld Package Management Guide*.

OneWorld makes certain assumptions about path code, foundation, and destination for the business function build. In particular, consider the following points:

- When building business functions, the path code you are building from is the path code you defined in the package.
 - The foundation is either the same as the foundation included in the package, or, for an update package, it is the associated foundation.
 - The destination where build output is directed is in the bin32, obj, and lib32 directories of the package itself.
- When building a full or partial package or an update package that includes a business function, you should always turn on the Build Business Functions option. Otherwise, the consolidated DLLs included in the package will not be current.

For update packages, a single build is performed for each business function defined in the package. After the individual business function is built, a global link is performed for that object and all other objects that are in the same consolidated DLL. The global link process uses the objects found in the check-in location for that package’s path code.

Similarly, any business functions defined in a partial package are built individually and then linked with all objects in the check-in location. The only difference with a partial package is that all consolidated DLLs are delivered, not just the ones that may be defined in the package itself.

Check-in location directories are not updated with the build’s output. For more information about the BusBuild program, see “Business Functions” in the *OneWorld Development Tools Guide*.

Before you begin, complete these tasks:

1. Assemble your package, making sure to set the assembled package's status to Definition Complete.
2. Verify that Object Configuration Management (OCM) mappings are correctly set for the Package Build Report (R9621) and Server Package Build Report (R9622) that are generated as part of the package build process. These reports need to be mapped to run locally.

13.6.10 Defining a package build

There are two ways to access the Package Build Definition Director: from the Package Assembly function's Work With Packages form or from the Package Build function's Work with Package Build Definition form:

- To access the Package Build Definition Director from the Work With packages form, select a defined package that has a Definition Complete Status, and then choose **Build Director** from the Row menu.
- To access the directory from the Package and Deployment Tools menu (GH9083), choose **Package Build** (P9621). The Work With Package Build Definition form appears.

To define a package build, perform the following steps:

1. Click **Add** to launch the Package Build Definition Director. The Package Build Definition Director form appears (Figure 153).

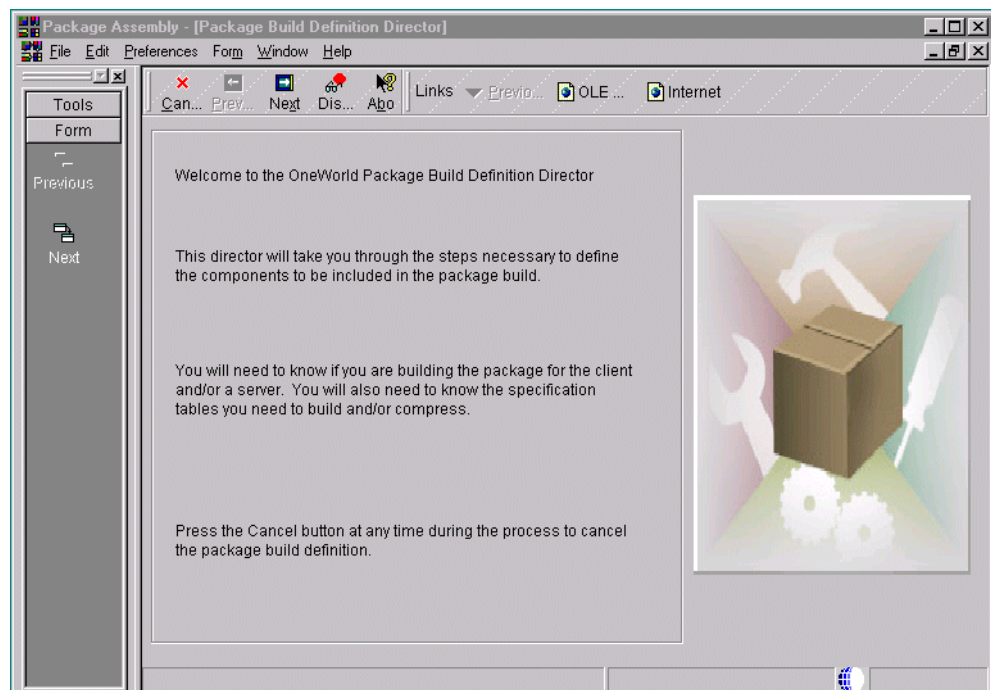


Figure 153. Package Build Definition Director

2. If you accessed the director by clicking Add, click **Next** on the Package Build Definition Director form to display the Package Selection form.

On the Package Selection form, find and choose the defined package you want to build.

If the package's definition is still In Definition, before you can build the package, you must change the status to Definition Complete. To do so, choose the package and choose **Activate** from the Row menu.

3. Click **Next**. The Package Build Location form appears (Figure 154). Specify whether you want to build a package for client workstations, one or more servers, or both clients and servers. If you are building a partial package, you will not be allowed to build the package for a server.

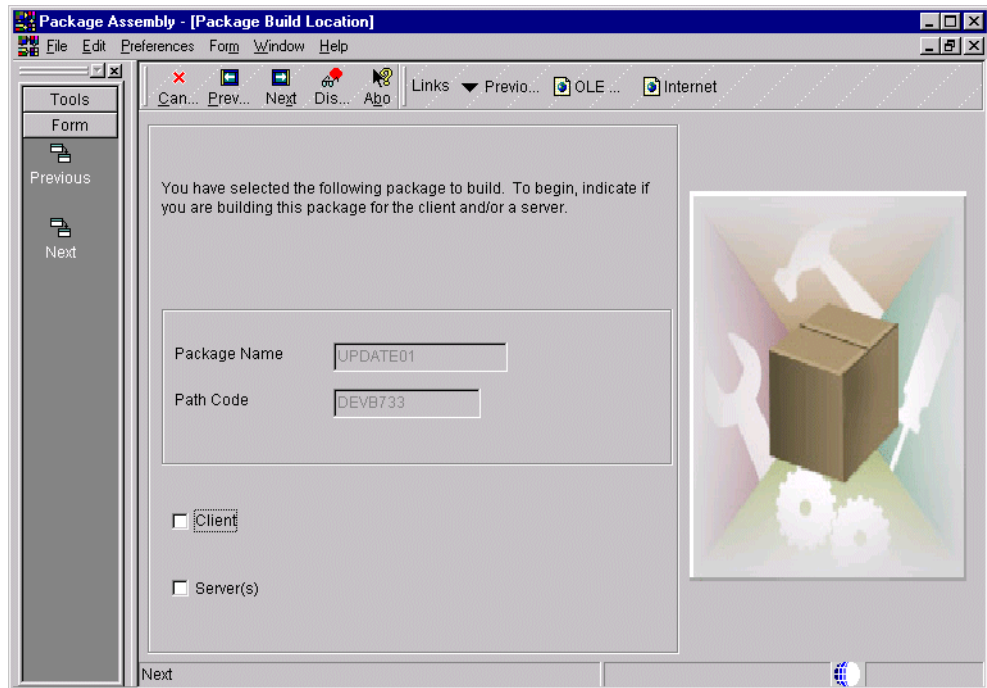


Figure 154. Package Build Location form

Be sure to click **Server(s)** if the package you are building goes to one or more servers.

If you are building a package for workstations only, skip the following step, which applies only to server package builds.

4. Click **Next**. If you are building a package for a server and you clicked Server(s) on the Package Build Location form, the Server Selection form appears.

This form enables you to choose the servers on which you want to build the package. To choose a server, click **Find**, and double-click in the column to the left of the server name. A check mark indicates your selection. Repeat the process to choose multiple servers.

5. Click **Next**. The Build Specification Options form appears (Figure 155).

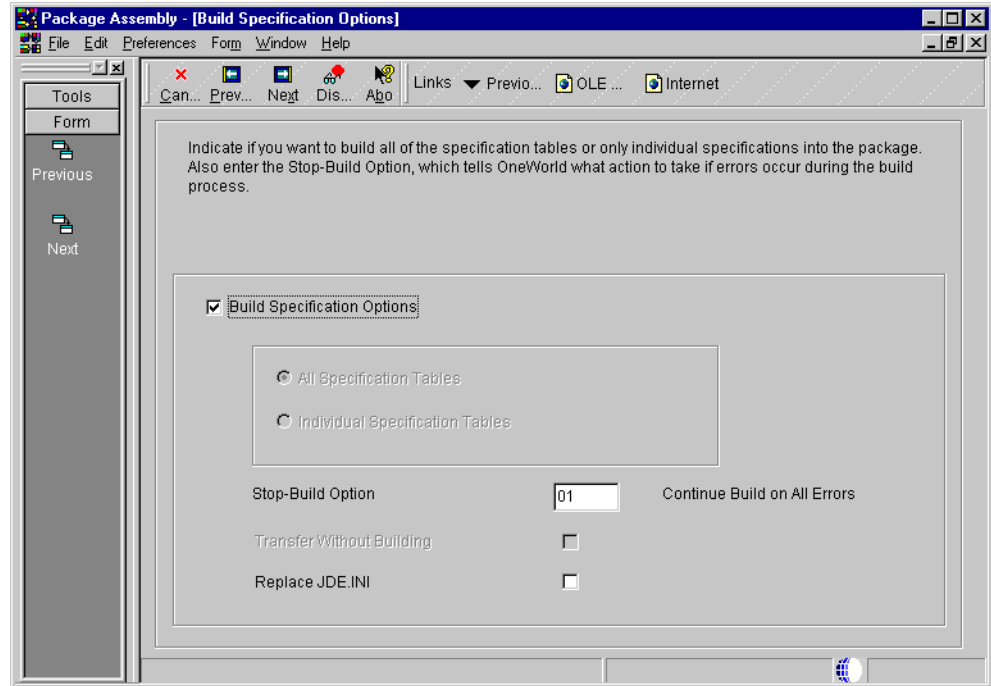


Figure 155. Build Specification Options form

Complete the following fields:

- Build Specification Options
- All Specification Tables or Individual Specification Tables
- Stop-Build Option
- Transfer Without Building (for server package builds only)
- Replace jde.ini (for update packages only)

For a full description of these fields, refer to the *OneWorld Package Management* manual.

6. Click **Next**. If you elected to build individual specification tables, the Individual Specification Selection form appears.
7. Click each specification table you want to build.
8. Click **Next**. The Business Function Options form appears (Figure 156 on page 358).

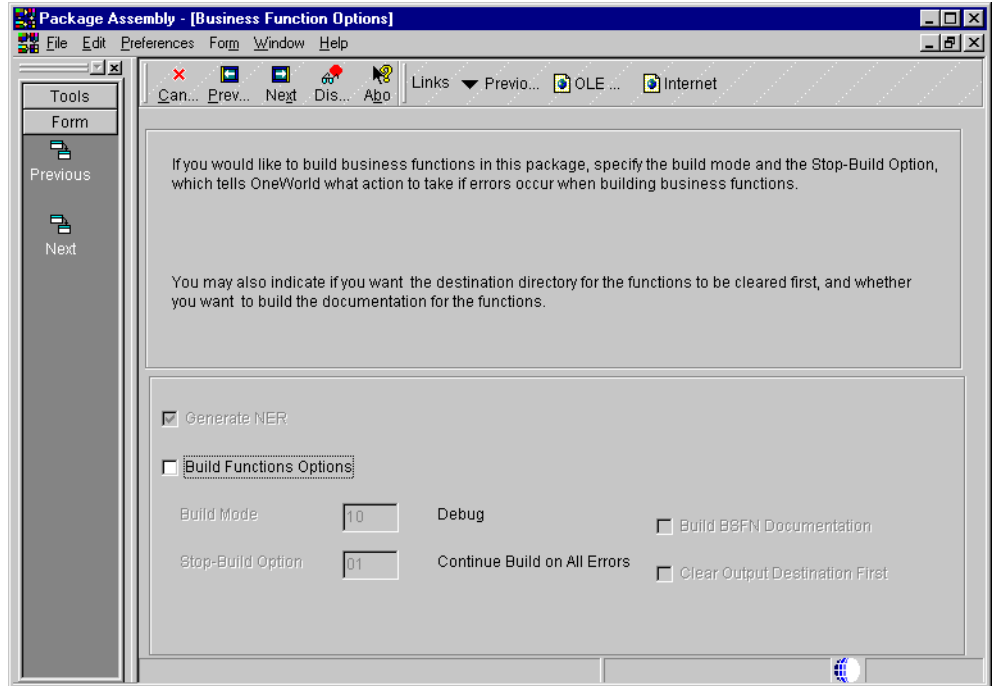


Figure 156. Business Function Options form

Complete the following fields:

- Build Functions Options
- Build Mode
- Stop-Build Option
- Build BSFN Documentation
- Clear Output Destination First

For a full description of these fields, refer to the *OneWorld Package Management* manual.

9. Click **Next**. The Compression Options form appears (Figure 157).

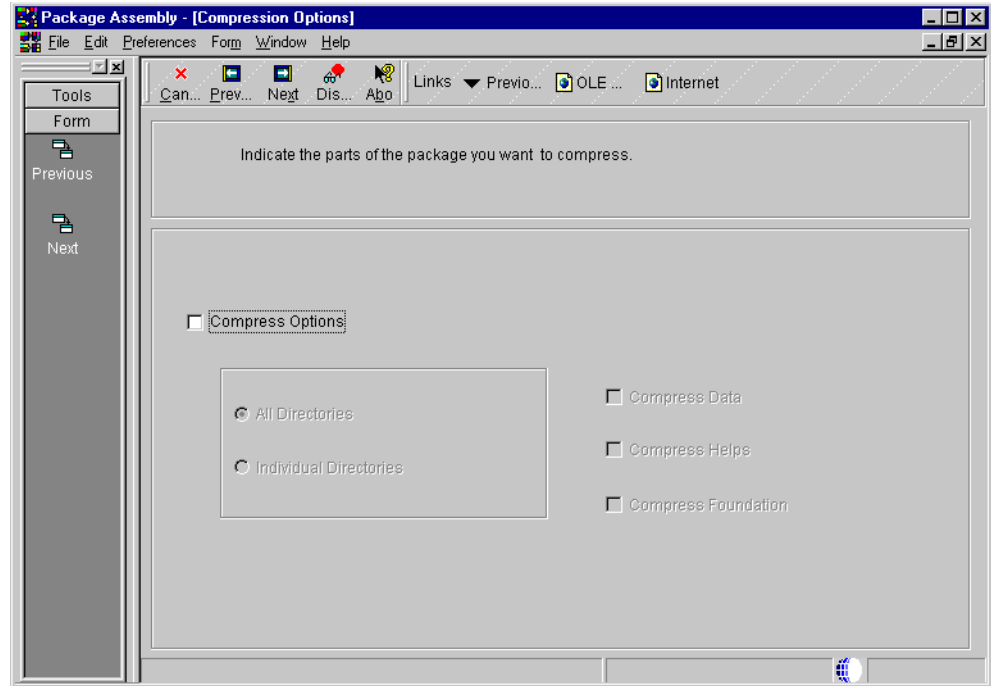


Figure 157. Compression Options form

Complete the following fields:

- Compress Options
- All Directories or Individual Directories
- Compress Data
- Compress Helps
- Compress Foundation

For a full description of these fields, refer to the *OneWorld Package Management* manual.

10. Click **Next**. If you elected to compress individual directories, the Individual Directory Selection form appears.
11. Click each directory that you want to compress.
12. Click **Next**. The Package Build Revisions form appears (Figure 158 on page 360).

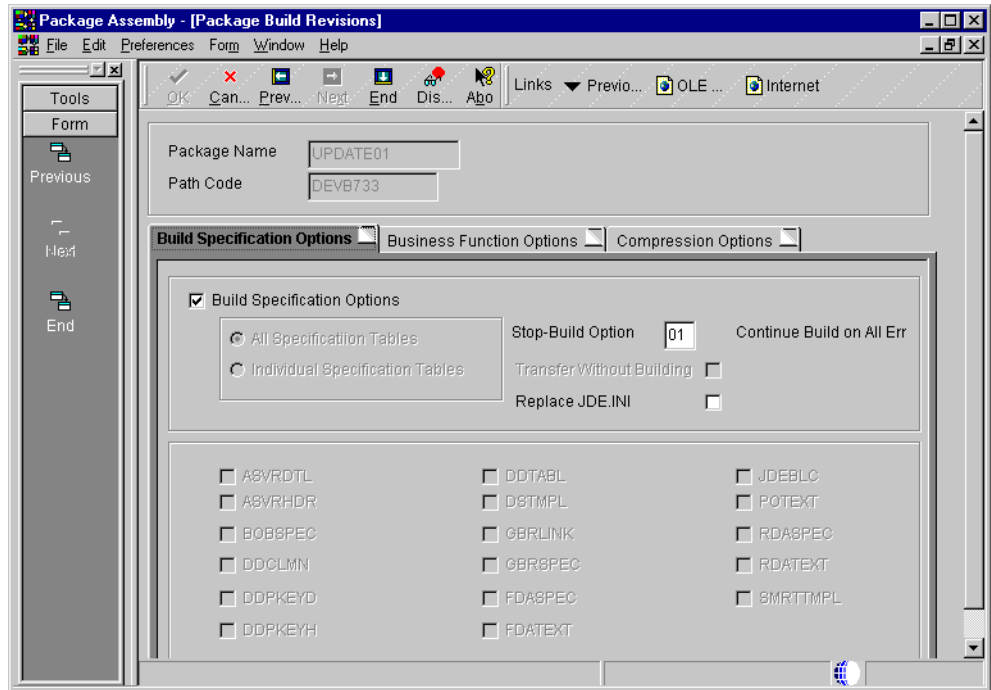


Figure 158. Package Build Revisions form

13. When you are finished reviewing or changing your build options, click **End** to exit from the Package Build Definition Director. The Work With Package Build Definition form appears.
14. When you are ready to initiate the package, highlight the package name, and select **Active/Deactivate** from the Row menu. Then choose **Submit Build** from the Row menu. Choose a destination for the build report, and click **OK**.
15. Review the report to make sure that all components in your package were built successfully. If the report indicates any errors, review the error logs for more detail.

After the package is successfully built, you must activate the package from the deployment application. Activating the package makes it available for installation through the Workstation Installation application. If you do not activate the package, the package will not be included in the list of available packages that displays when Workstation Installation is launched.

If the package build completes successfully, you can schedule the package for deployment as described in the *Deployment* section of the *OneWorld Package Management Guide*, which also includes information on the following items:

- Revising build options for a package
- Copying a build package
- Viewing a package build history and resubmitting builds

Table 23 describes in greater detail the fields displayed while defining the package build.

Table 23. Field names and explanations

Field	Explanation
SAR Number	An abbreviation for Software Action Request (SAR). This number is used by the J.D. Edwards project control system to keep track of all charges and activity by individual SAR number.
Object Name	<p>The OneWorld architecture is object based. This means that discrete software objects are the building blocks for all applications, and that developers can reuse the objects in multiple applications. Each object is stored in the Object Librarian. Examples of OneWorld objects include:</p> <ul style="list-style-type: none"> - Batch Applications - Interactive Applications - Business Views - Business Functions - Business Functions Data Structures - Event Rules - Media Object Data Structures
Description	<p>The description of a record in the Software Versions Repository file. The member description is consistent with the base member description.</p> <p>..... <i>Form-specific information</i></p> <p>On this form, a description of the object</p>
Object Type	The type of object with which you are working. For example, if you are working with tables, the object type is TBLE, or business function is BSFN.
Status Code	This code determines the status of the software in the development cycle.
User	For World, the IBM-defined user profile. For OneWorld, the identification code for a user profile.
Merge Option	The Merge Option denotes whether a customer's OneWorld object will be merged in with the J.D. Edwards OneWorld object. The Merge Option can be set at the Path Code level so that all objects checked into that path will carry the same Merge Option as the Path Code.
Date Modified	The Date Modified field is the date that the DREAM Writer version, Software Versions Repository Record, and so on, was last updated.
Time Modified	The time the object was last checked in.
Location	For World, the Location indicates the machine (server or client). For OneWorld, the Location or Machine Key indicates the name of the machine on the network (server or workstation).

Field	Explanation
Package Name	<p>For World, name identifying an installation package. For OneWorld, a package describes the location on the server where components you want to deploy to workstations reside. There are three package types:</p> <p><i>Full</i>—Contains the full suite of OneWorld applications (all specifications). <i>Partial</i>—Consists of a minimum configuration of OneWorld. <i>Update</i>—Contains new or modified objects.</p> <p>With the exception of just-in-time applications included in an Update package, all packages are a “snap shot” at a point in time of the central objects for a particular path code. Just-in-time applications are dynamic, not built.</p>
Package Type	<p>Full and partial packages are primarily for initial OneWorld installations, and allow you to create a full or partial workstation configuration on machines that do not have OneWorld installed. These packages are normally deployed using the Workstation Installation application. Update packages are for deploying changes, additions, or enhancements to users who already have OneWorld installed, or can be used as a complete replacement of an existing package. Update packages are always installed through Package Deployment Instructions.</p> <p><i>Full package</i>—This is a full package that contains all specifications and foundation code needed to run and develop in OneWorld.</p> <p><i>Partial package</i>—This package type is a minimum configuration of OneWorld that contains only the specifications that allow users to launch OneWorld Explorer. Applications can then be loaded through Just-In-Time-installation the first time the user selects an application.</p> <p><i>Update Package</i>—This package type enables the OneWorld administrator to create a package containing specific items which can then be deployed to specific users. An update package enables administrators to include in the package any item type (such as any OneWorld object, help file, OneWorld foundation, and so on). For this reason, this package type is well-suited for deploying software changes and fixes quickly. When a user is scheduled to receive an update package, OneWorld loads the package the next time the user signs on to OneWorld.</p>
Package Build Status	The status of the current build for the package.
Build Mode	There are three modes: Debug, Optimize, and Performance. Debug and Performance are for JD Edwards developers only. Users should select Optimize mode.
Parent Package	Because an update package includes only a subset of objects, you must indicate the parent package on which the update package is based or to which is related. This information is used by the system to determine how to build business functions.

Field	Explanation
Foundation Name	A OneWorld foundation is the code required to run all OneWorld applications. A foundation ID is required for all full and partial packages. This is normally the system directory at the same directory level as your path code. The foundation must be compressed when built.
Service Pack Number for OW Foundation	A service pack is an update to the foundation code that is delivered between major releases and cumulative releases of OneWorld software.
Include Object Specs	When you include individual objects in an update package, you have the option of including a corresponding set of specifications for those objects. When you include specifications, a “snapshot” of the specifications will be included in the package after it is built. When the package is deployed, the package recipient receives those object specifications. If you do not include object specifications in the package, the old specifications for the objects in the package will be deleted from the workstation when the package is deployed. Then, the next time the package recipient attempts to use the object, a new set of specifications for it will be transferred to the workstation via Just-In-Time Installation.
Item Name	<p>The OneWorld architecture is object based. This means that discrete software objects are the building blocks for all applications, and that developers can reuse the objects in multiple applications. Each object is stored in the Object Librarian. Examples of OneWorld objects include:</p> <ul style="list-style-type: none"> - Batch Applications - Interactive Applications - Business Views Business Functions - Business Functions - Data Structures Event Rules Media - Object Data Structures <p>.....<i>Form-specific information</i></p> <p>On this form, this is the name of the OneWorld object you want to include in the package.</p>
Path Code	<p>The Path Code is a pointer to a set of OneWorld objects, and is used to keep track of sets of objects and their locations within OneWorld.</p> <p>.....<i>Form-specific information</i></p> <p>On this form, the path code refers to the set of central objects on which this package is based.</p>
Replace jde.ini	Indicate if you want a new jde.ini file delivered with the package. You should leave this unchecked unless your jde.ini file has changed. For example, your jde.ini may change when you perform upgrades or when you reconfigure in release master.
JDE Versions	Indicate if just the JDE versions (XJDE and ZJDE) for the UBE selected should be included in the package

Field	Explanation
Version	A version is a user-defined set of specifications. These specifications control how applications and reports run. You use versions to group and save a set of user-defined processing option values and data selection and sequencing options. Interactive versions are associated with applications (usually as a menu selection). Batch versions are associated with batch jobs or reports. To run a batch process, you must choose a version.
Server Package Name	An abbreviation for Software Action Request (SAR). This number is used by the JDE project control system to keep track of all charges and activity by individual SAR number.
Build Package	<p>The Yes or No Entry field is a common single character entry field for simple yes or no responses on prompt screens.</p> <p>.....<i>Form-specific information</i>.....</p> <p>Build Package takes the package definition and copies and converts the object from the Central Objects data source to the replicated format used by workstations.</p> <p>Build Business Functions performs a global build of all business functions included in this package.</p> <p>Compress Package compresses the application objects included in the package. It also creates a package INF that indicates that the following items are compressed: foundation, help, data and application objects. The compress package option is the only one that modifies the package INF. All other compression options only perform the actual compression.</p> <p>Compress Foundation compresses the foundation associated with this package.</p> <p>Compress Data compresses the Access database associated with this package.</p> <p>Compress Helps compresses the help files associated with this package.</p>
Build Individual Specifications options	On this form, check one or more individual specification files to build in the package. You should use this feature only if your package build failed and the package error log indicates that an individual specification file needs to be rebuilt. This saves you time when you want to build only one or more individual specification files rather than rebuilding the entire package.
Compress Individual Directories options	On this form, check one or more individual directories to compress. You may use this if you previously compressed the package, and need to add files to a directory and recompress. This compression option does not update the package INF file. It is simply used to save time by compressing one or more package directories rather than the entire package.
Build Mode	Indicate the compiler configuration to use for the software build.

Field	Explanation
Treat Warnings As Errors	When building business functions, you have the option of reporting all compiler warnings as errors or as warnings. The consolidated DLL for a function with a warning or error will still be built. However, if the function has an error, that function will not be built into the consolidated DLL. Functions with warnings will be built into the consolidated DLL.
Clear Output Destination Before Build	When building business functions, you should normally clear the output directories (bin32, lib32, and obj) so that old business functions are not included when the package is created. If you do not clear the output and a function fails to process, the global link process uses the old libraries and objects to create the consolidated DLL file. This can result in an inaccurate global package.

Refer to the *OneWorld Package Management* manual for an explanation of fields not included here.

13.6.11 Tables used in a package build

Package build is based on tables created by the Package Build application. The following tables are used in a package build:

- F9603: Header Table
- F96021: Package Build Table
- F96215: Package Build History Table
- F9631: Package Detail Table (replaced F9886)
- F9622: Build Detail Table
- F96225: Build Detail History Table

13.6.12 Verifying package build errors

After you build your package, you can view logs that list any errors that occurred during the build process. In particular, you can view the following logs:

- Package Statistics Log
- Package Build Log
- Business Function Errors Log
- Missing Business Function Source Errors Log

Each log contains a header, which includes the package name, date, build machine, and path code.

To view a log, perform the following steps:

1. On the Package Build menu (GH9083), choose **Package Build History** (P9622). The Work With Package Build History form appears (Figure 159 on page 366).

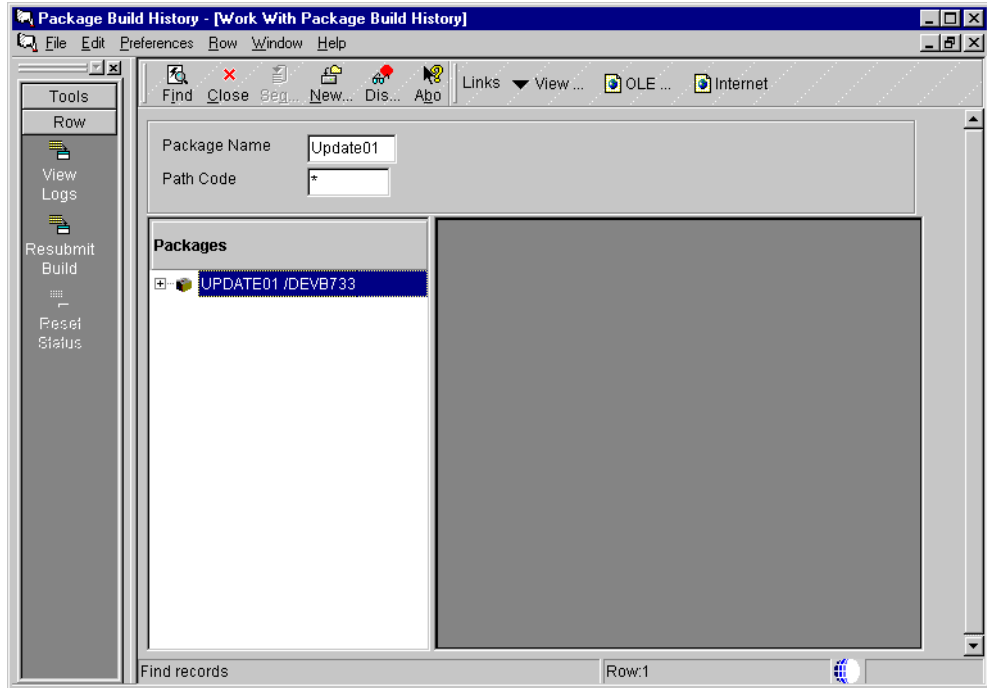


Figure 159. Work With Package Build History

2. Choose **View Logs** from the Form menu. The View Logs form appears (Figure 160).

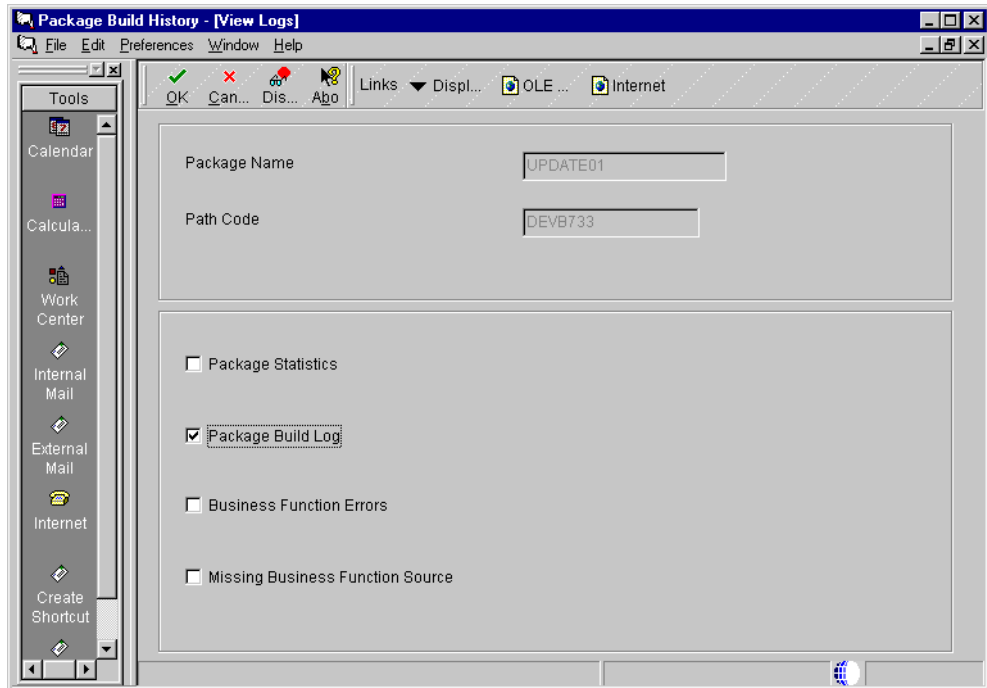


Figure 160. View Logs

3. Click any of the following log options:
 - Package Statistics
 - Package Build Log

- Business Function Errors
- Missing Business Function Source

Each log you selected displays in its own window. Click **OK**.

4. When you are finished with view logs, close each log window.
5. From the View Logs form, click **Cancel** to return to the Work with Package Build History form.

You can view error logs without using the Package Build History application by locating the desired log in the correct directory. Error logs are stored on the Deployment Server in directories beneath the package itself. You can view the error logs by going to the appropriate directory and opening the log with Microsoft's Notepad or a similar application that allows you to display text files. Table 24 shows the various logs that are generated during the package build process and their locations.

Table 24. Log generated during the package build

Type of log	Description	Where to find the log
Package Statistics	This option allows you to view count and size statistics for the package directories that were built.	\\packagename\work\buildreport.log
Package Build Log	This option allows you to view errors that may have occurred during a package build. These errors could have occurred while building the specification files or the objects for the package.	\\packagename\builderror.log
Business Function Errors	This option allows you to view the results of the business function build for this package. Both errors and warnings display in this report. A summary appears at the end of the report that indicates how many errors and warnings occurred for each DLL. Use this information to determine if a rebuild is necessary.	\\packagename\work\buildlog.txt
Missing Business Function Source	This option lists all source members that were not available when the business function was created. The program attempted to find these members because each had a record in the Object Librarian table (F9860). However, a matching source could not be found in the source directory. To resolve these errors, either delete the Object Librarian record or provide a source member.	\\packagename\work\NoSource.txt

13.6.12.1 The Package Statistics log

The Package Statistics log summarizes the outcome of the package build by showing statistics for the directories in the package, including the size and file count of each directory. This report also shows a breakdown of the files in the Spec directory and the size of each Spec file, as well as the total count and size. The purpose of this report is not only to provide information, but to verify that the package was built successfully.

13.6.12.2 The Package Build log

The Package Build log can be found in the package name directory. This log lists the steps executed in building the package, as well as any errors that occurred during the package build process. It also describes the steps involved in building the package. The final page of the log tells you whether the package was successfully built. Error and warning messages are preceded with the words ERROR and WARNING respectively. All other messages are informational

messages that describe the steps in the build process. If the build process fails, you can use this log as a troubleshooting tool to determine the last step that executed before the failure.

For example, the following steps are from the log. They include some troubleshooting suggestions to follow if the build fails during or after the step:

1. **Build Directories for Package Complete:** A failure at this point means that the directories in the package cannot be built. If this occurs, make sure the server did not go down. Also, make sure you have authority to create directories in the package directory.
2. **Build Package Definition List:** This step builds a list of all items defined in the package. If the build fails at this step, make sure the package is defined in the F9631 table.
3. **Create Empty TAM specs:** This step creates the local workstation specifications (TAM tables) in the package directory under the specs directory. If you get a failure here, make sure the server did not go down. Also, make sure you have permission to create directories in the Spec directory.
4. **Copy lib32, source, include, obj, make and work for [Full] package:** This step copies the lib32, source, include, obj, make, and work directories from the check-in location to the package directory. If you receive a failure at this point, make sure that each of these directories exist in the check-in location and under the package directory.
5. **Copy lib32, make, and work for Partial package:** This step copies the lib32, make, and work directories from the check-in location to the package directory. If you receive a failure at this point, make sure that each of these directories exist in the check-in location and under the package directory.
6. **Spec file DDCLMN begun, Spec file DDCLMN finished:** These two messages go together. If you receive the *begun* message but not the *finished* message, this means the Spec file was not built. After the package is finished, you need to build just the one TAM Spec file that failed. There may be an entry in the log that there was an error building that TAM Spec file.
7. **The Business Functions Errors Log:** The Business Functions Errors log allows you to view any errors that occur while business functions are being built. The final page of the log describes whether the business functions were successfully built or built with errors. Business functions in this report may be business functions that are still in development and are not yet checked in. If they were never checked in, they will not have a source and, therefore, will be listed in the missing source log.

13.6.12.3 The Missing Business Function Source Errors log

The Missing Business Function Source Errors log describes any business functions in the package that are defined in the Object Librarian and have a record, but cannot be built because there was no source.

13.6.12.4 Post-build considerations

In certain situations where you make business function changes between package builds, the latest changes may be reflected in the original package, but not in the new package. When you build a full or update package, follow the recommended strategies after your package is built to ensure that your package recipients receive all changes to business functions.

13.6.13 Strategy for full packages

When you build a full package, the DLL, LIB, and OBJ files reside in the \<packagename> directories. The next time you create a full package, objects are pulled from the BASE path code. This means that if you make any business function changes between the time you built the original full package and the new package, the new DLL and LIB files will reside only in the original full package. The new package will not contain any of the changes you made after the original full package was built.

The solution is to manually copy files from the machine where the package was created to the base path code directory. Your directory structure may look similar to the display in Figure 152 on page 353.

Copying files for full packages

To copy files for full packages, complete the following steps:

1. Copy all of the DLL, LIB, and OBJ files from the <path_code>\<package> directory, on the machine where the package was initially created, to the base path code directory.
2. Copy to the base path code directory any files from the source and include directories that were newly created when generating named event rules (NER). To determine which files to copy, in the Windows Explorer window click the Modified heading bar to sort according to the date that the files were last modified. There should be 10,000 or more files in each directory. Replace the \bin32, \obj, and \lib32 directories under \prodb733 with the corresponding directories under \<packagename>.
3. Copy the files in the \include and \source directories under \<packagename> into the corresponding prodb733 directories.

13.6.14 Strategy for update packages

When you build an update package that contains business functions, the DLL and LIB files for that business function reside in the package directory on the Deployment Server after the business function is compiled. If you build another update package that contains a business function with the same parent DLL as the original business function, the DLL that is created does not contain the previous package's business function. When you deploy the new package, the original update package's business function is overwritten by the new one. To avoid this situation, use only one cumulative update package for all business functions. This means that you should add business functions into the existing update package. As an alternative, you can keep track of all previous business functions and include them in a new update package to ensure that all previous business functions are not overwritten.

Refer to the *OneWorld Package Management* manual for information on:

- Resubmitting a package build
- Changing the build status
- Understanding package INF files

13.6.15 Understanding package deployment

After you build a package, there are several methods available for deploying the package to workstations and servers throughout your enterprise. For

workstations, the method you select depends on whether OneWorld is already installed on the workstation.

13.6.15.1 Deployment to workstations without OneWorld

If a workstation does not currently have OneWorld installed, you can deploy the package through the Workstation Installation program. Workstation installation is used to deploy only full and partial packages. You cannot use Workstation Installation to deploy an update package to a workstation that doesn't have OneWorld installed.

Workstation Installation retrieves from the Central Objects data source the items specified in the package. A package is like a bill of materials with instructions describing from where to pull all of the necessary components that the Workstation Installation program deploys to the local workstation. This program can be run interactively (initiated by a person at a workstation) or in silent mode and scheduled through the push installation feature. For more information about the Workstation Installation application, see *the OneWorld Installation Guide*.

You can use Package Deployment to deploy the package if you use the push installation feature. Push installation enables the OneWorld administrator to initiate the installation of a package from the Deployment Server to workstations without requiring any user interaction. To use this feature, the push installation "listener" application must be installed on the workstation, and the machine must be defined through the Machine Identification application (P9654A). For more information about push installation, see the *OneWorld Package Management Guide*.

13.6.15.2 Deployment to workstations with OneWorld already installed

To reload a new package (full, partial, or update) on workstations that already have OneWorld installed, use one of the two methods: Workstation Installation (refer to 13.3.1, "Workstation Installation" on page 333) or Package Deployment (refer to 13.3.2, "Package Deployment Instructions" on page 333).

After you assemble and build a package, use Package Deployment to schedule the package for deployment to individual workstations or to selected groups. On the specified deployment date, when your users scheduled to receive the package sign on to OneWorld, they are given the opportunity to load the package.

Unless you are using the push installation feature, Package Deployment requires that OneWorld be already loaded on the workstation. You can schedule either a new full or partial package to replace the existing package, or an update package to be merged with the existing package on the workstation.

There are advantages to both deployment methods. Workstation Installation is a good method to use when you want to install a package immediately or soon after it is built, without having to schedule the package. On the other hand, Package Deployment is useful if you need to control when the package becomes available, if you want to make the package installation mandatory, or if you want to deploy the package to servers as well as workstations.

13.6.15.3 Deployment to servers

Servers receive the same package you build for the workstation, but in a different format. When you assemble the package and create the package build definition, you can specify the servers to which you want to deploy the package.

Deployment is accomplished via the Package Deployment application, which uses the same scheduling mechanism used to deploy packages to workstations. In fact, you can easily schedule deployment to both client workstations and servers on the same form. You cannot use the push installation feature for deploying servers.

13.6.15.4 Deployment to tiered deployment locations

Multi-Tier Deployment allows you to install software on workstations from more than one deployment location and more than one deployment machine. Use this deployment method if your site has more than 50 workstations performing OneWorld software installations per day, or when workstation installations over your WAN are too slow. For more information about Multi-Tier Deployment, see the *OneWorld Package Management Guide*.

13.6.15.5 Deploying to workstations from CD

If your system has a CD writer, you can define the CD writer as a deployment location. Doing this essentially defines the CD writer as a pseudo Deployment Server from which you can copy a OneWorld package onto a blank CD. You can then use this CD to install OneWorld on workstations by using the Workstation Installation program contained on the CD.

For more information about deploying from CD, see the *OneWorld Package Management Guide*.

13.6.16 Working with the Package Deployment Director

After you define and build a package, use Package Deployment to schedule the package for deployment to individual workstations, Deployment Servers, or Enterprise Servers. On the specified deployment date, users who are scheduled to receive the package are given the opportunity to load the package when they sign on to OneWorld.

Alternatively, you can schedule the package to deployment groups or locations instead of specific machines. Deployment groups are useful in large enterprises that routinely deploy packages to many workstations and servers.

13.6.16.1 Understanding the Package Deployment Director

The Package Deployment Director is designed to simplify and expedite the process of scheduling built packages to workstations and servers. The director displays a series of forms that allow you to specify the package you want to deploy, the deployment destinations, and the deployment time. After specifying the package you want to deploy, specify any of the following destinations:

- Client workstation
- Enterprise server
- Deployment server

Or, you can choose to specify one of these destinations:

- Deployment groups
- Locations

You can deploy a package either to specific workstations and servers, or you can schedule the deployment based on deployment groups or location. You cannot do both. You must choose one method or the other.

You can make the package mandatory, which means users cannot access OneWorld until they have installed the package. If the package is optional, users will be given the option of installing the package every time they sign on to OneWorld until they either install or decline the package.

In addition, you can specify a “push installation”, which means the package can be pushed from the Deployment Server to the workstations you specify, without requiring any interaction from the user. For more information on push installation, see the *OneWorld Package Management Guide*.

The Package Deployment Director requires that OneWorld be already loaded on the workstation, unless you are using push installation. You can schedule either a new full or partial package to replace the existing package or an update package to be merged with the existing package on the workstation.

The Package Deployment Director uses the following tables:

- F9650 Machine Master
- F9651 Machine Detail
- F9652 Deployment Group Header
- F9653 Deployment Group Detail Definitions
- F9654 Deployment Location Definitions
- F98825 Package Deployment Information
- F968826 Package Deployment on Servers
- F9603 Software Package Header

13.6.16.2 Using the Package Deployment Director

Once you have assembled and built your package, defined all machines, and verified your deployment groups, you are ready to use the Package Deployment Director to specify package recipients and schedule the package for deployment.

When you schedule a package for deployment to a machine rather than a deployment group or location, you can schedule to deploy the package to client workstations, Deployment Servers, Enterprise Servers, or a combination. Depending on your selection, different forms appear so you can enter information pertaining to your selection. For example, if you indicate that you want to schedule a package for deployment to client workstations and a Deployment Server, the forms for selecting specific workstations and Deployment Servers appear. If you schedule a package for deployment to only client workstations, the server selection form does not appear.

To schedule a package for deployment to a client workstation or server, perform following steps.

1. From the Package and Deployment Tools menu (GH9083), choose **Package Deployment** (P9631). The Work with Package Deployment form appears (Figure 161).

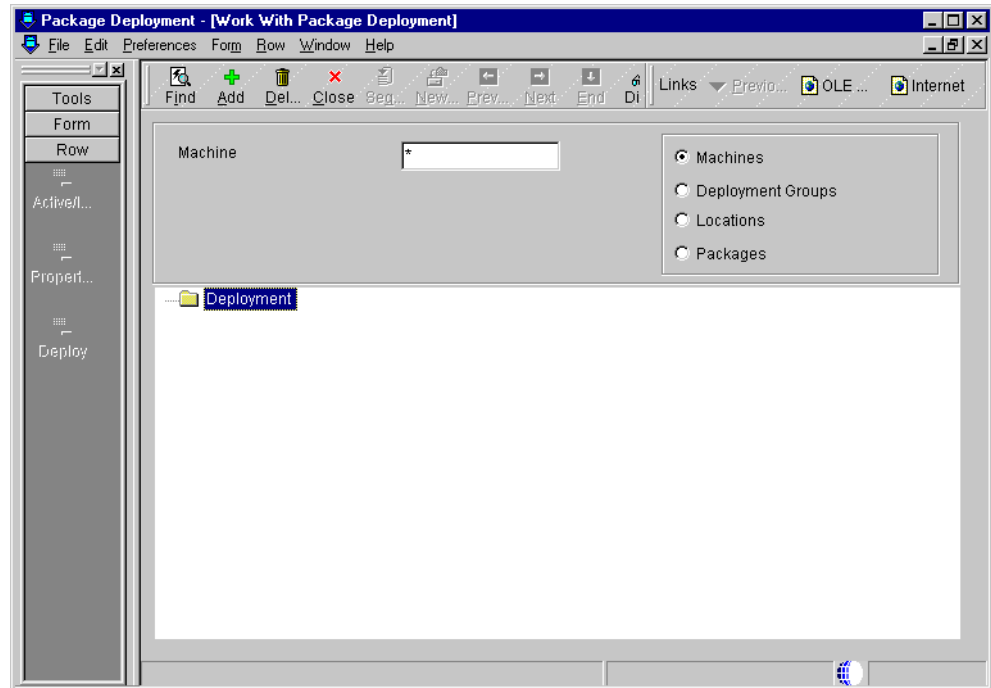


Figure 161. Work With Package Deployment

2. Click **Add** to launch the Package Deployment Director. The Package Deployment Director form appears (Figure 162).

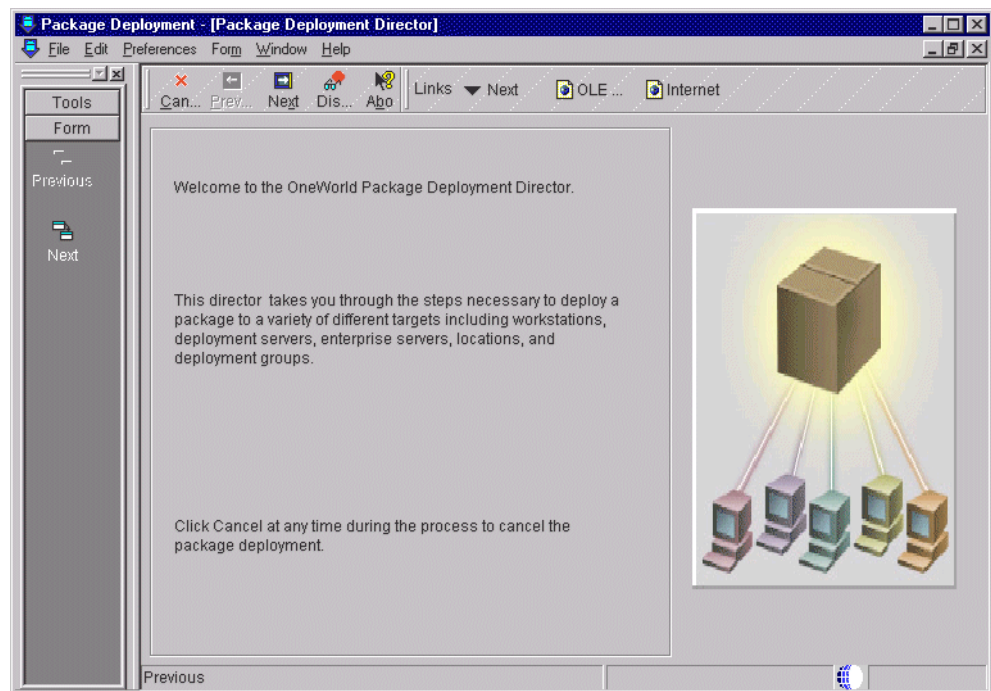


Figure 162. Package Deployment Director

3. Click **Next**. The Package Selection form appears (Figure 163 on page 374).

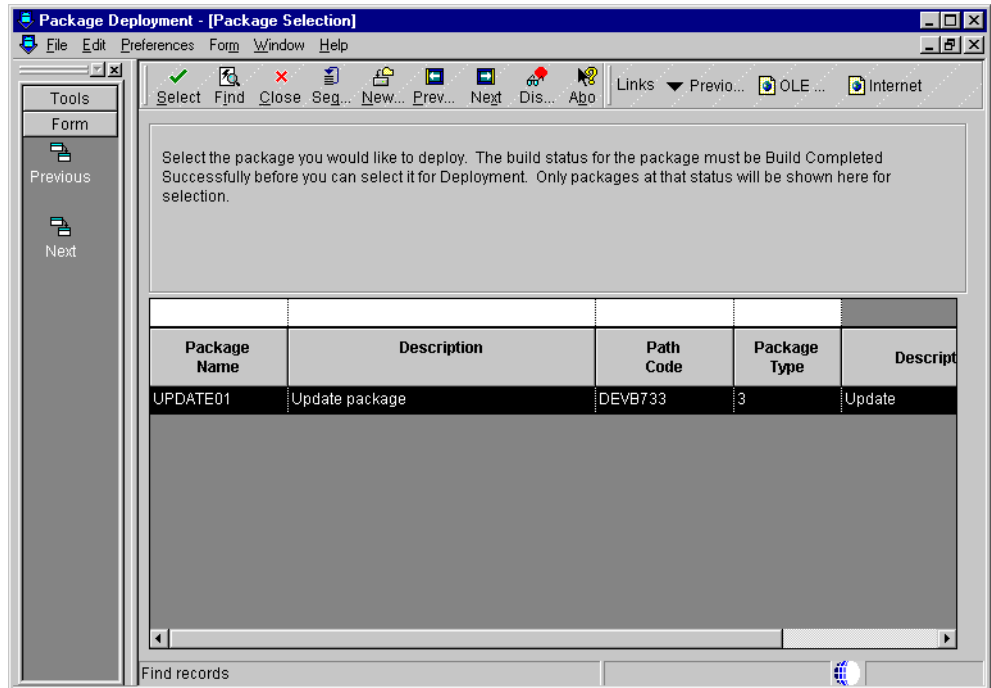


Figure 163. Package Selection

4. Choose the package you want to deploy. Click **Next**. The Package Deployment Targets form appears (Figure 164).

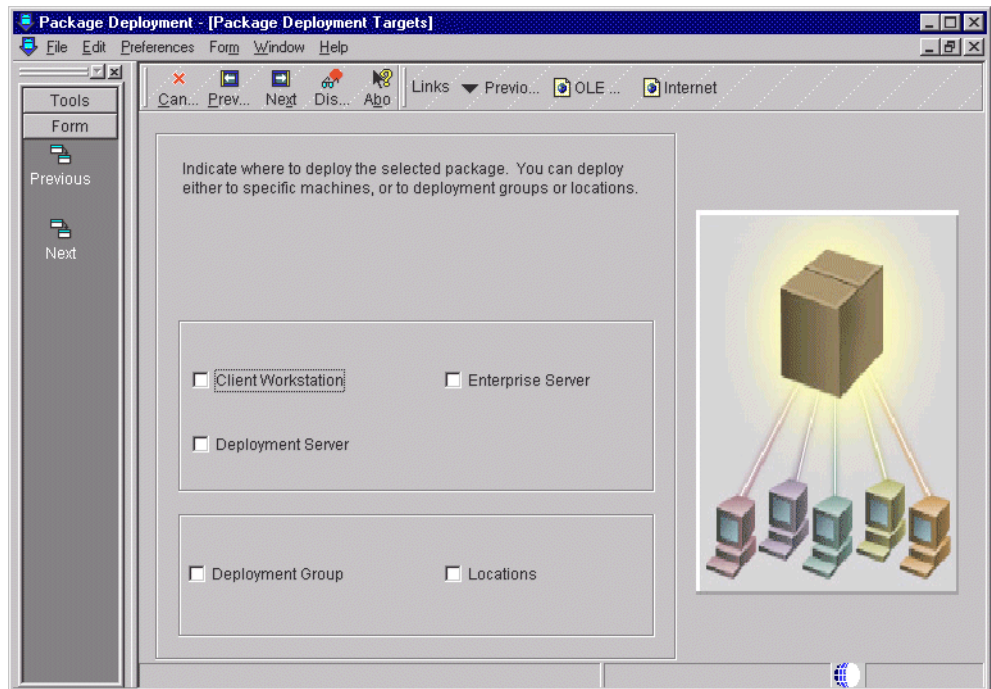


Figure 164. Package Deployment Targets

Indicate the type of machines to which you want to deploy your package:

- Client Workstations
- Deployment Servers
- Enterprise Servers

5. Click **Next**. The Package Deployment Attributes form appears (Figure 165).

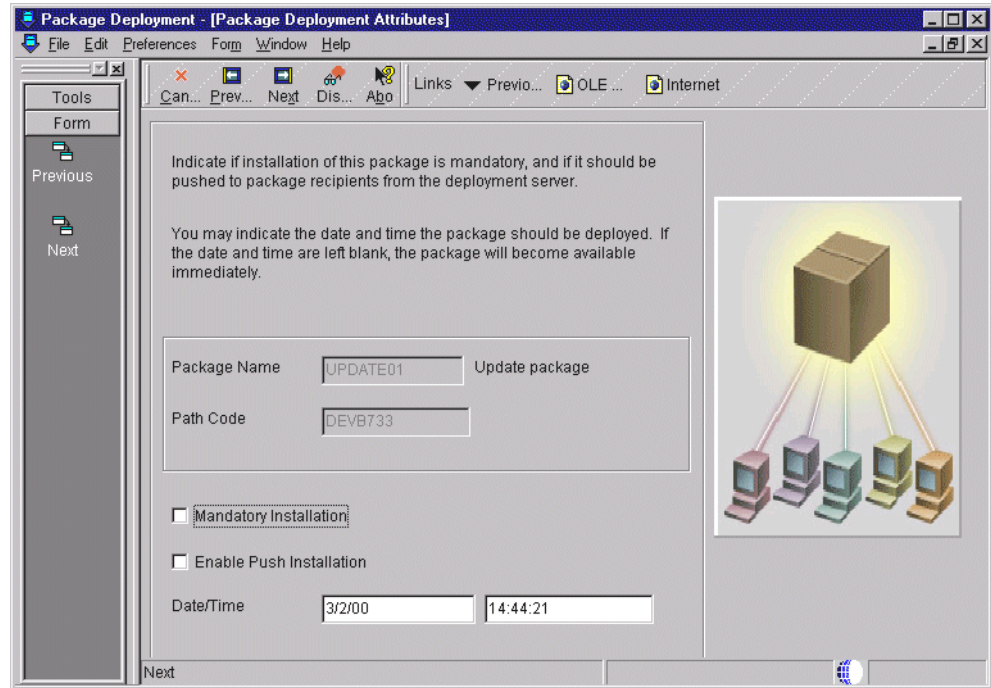


Figure 165. Package Deployment Attributes

Complete the following fields:

- Mandatory Installation
- Enable Push Installation
- Date
- Time

Click the **Enable Push Installation** option if you want to deploy the package using push installation, which “pushes” the package to workstations from the Deployment Server.

6. Click **Next**. If you are deploying to workstations, the Deployment Client Workstation Selection form appears (Figure 166 on page 376). If you are not deploying to workstations, skip this step.

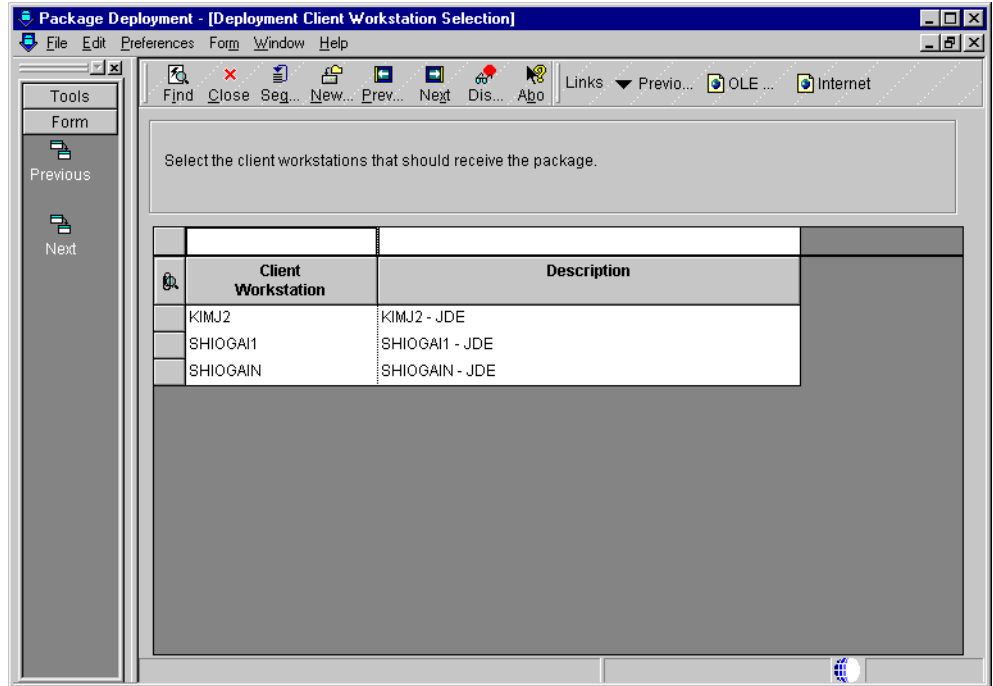


Figure 166. Deployment Client Workstation Selection

Find and choose the workstations to which you want to deploy the package. Choose a workstation by double-clicking in the column to the left of the Client Workstation column.

7. Click **Next**. If you are deploying to a Deployment Server, the Deployment Server Selection form appears. If you are not deploying to a Deployment Server, skip this step.

Find and choose the Deployment Server to which you want to deploy the package. Choose a server by double-clicking the column to the left of the Deployment Server Name column.

8. Click **Next**. If you are deploying to an Enterprise Server, the Enterprise Server Selection form appears. If you are not deploying to an Enterprise Server, skip this step.

Find and choose the Enterprise Server to which you want to deploy the package. Choose a server by double-clicking in the column to the left of the Enterprise Server Name column.

9. Click **Next**. The Work with Package Deployment form appears (Figure 167).

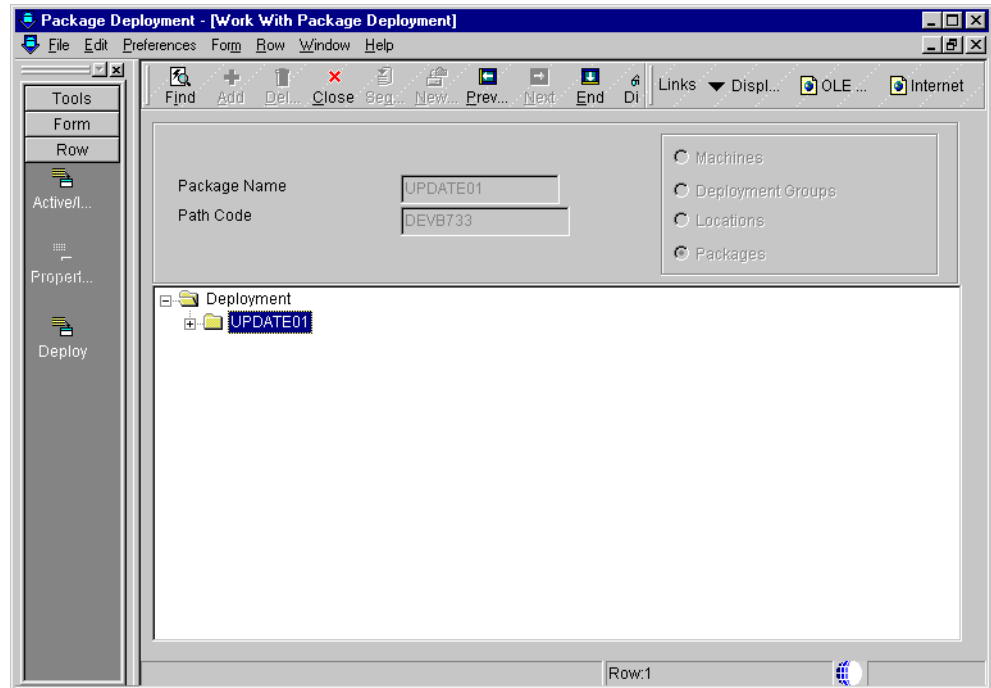


Figure 167. Work with Package Deployment

This form enables you to review your deployment selections. To change any of your selections, click **Prev** to return to the appropriate previous form.

10. When you are finished viewing or changing your deployment selections, click **End**.
11. If you are deploying a server package, you must find and choose the server package on the Work with Package Deployment form, and then choose **Deploy** from the Row menu. After you schedule your package for deployment, at the specified time on the date you specified, the package is deployed to workstations. This package will be made available when the user signs onto OneWorld.
12. If this is a full or partial client package that will be seen through the Install Manager, highlight the package name and select **Activate/Deactivate** from the Row menu. This sets the DeploymentStatus in the package.inf file to Approved. By doing this, the user can now see the package through the Install Manager.

Chapter 14. Printing

This chapter contains a general overview of AS/400 printing, as well as printing setup and issues specific to OneWorld.

14.1 AS/400 printing

This section explains components and capabilities of AS/400 print management. Major components of AS/400 printing system are:

- Printer device description
- Output queue
- Printer file definition
- Printer writer
- Spooled file

Figure 168 shows a flow of to-be-printed data from an AS/400 application to the printer.

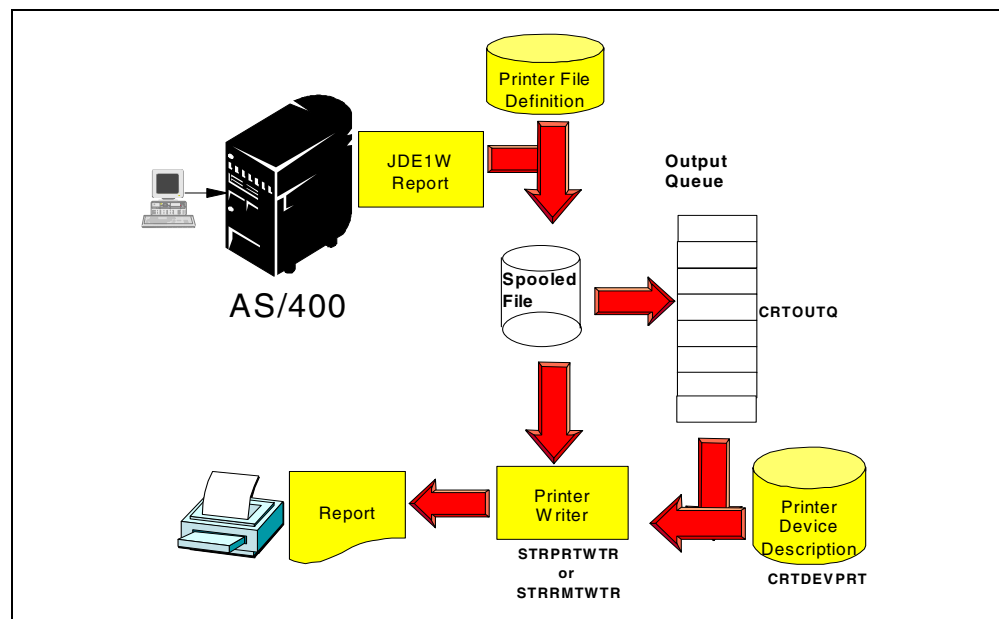


Figure 168. AS/400 printing process flow

An application creates a *report*, which is stored in a *spooled file*. The spooled file contains the to-be-printed data. Spooled file attributes (such as page size, number of copies, default font, and so on) are defined in the *printer file* associated with the application. The spooled file is placed into an *output queue*, where it waits until it will be printed. A *printer writer* program then processes the spooled file to generate the report on the printer. During this process, the printer writer takes information about the printer from the *printer device description*.

14.1.1 AS/400 printer attachments

This section shows the different printer attachment methods on the AS/400 system depending on the type of printer. The following attachment methods are a few of the possible alternatives:

- Printers attached to AS/400 workstation controller or to an IBM 5x94 (Remote Control Unit)
- LAN attached Intelligent Printer Data Stream (IPDS) printers
- ASCII printers attached to a PC
- LAN attached ASCII printers

Note that these are printers directly attached and controlled by an AS/400 system. In other words, these are printers for which there is a device description on AS/400 system. All printers attached to remote systems or connected using a TCP/IP LPR/LPD attachment are discussed in 14.1.7.2, “TCP/IP LPR and LPD printing” on page 386.

For more information on printing SCS, IPDS, AFPDS, or USERASCII spooled files on the different attachment methods and for information on IBM printers, refer to *AS/400 Printer Device Programming*, SC41-5713, and *IBM AS/400 Printing V*, SG24-2160.

Figure 169 shows some possibilities of attaching printers to AS/400 system.

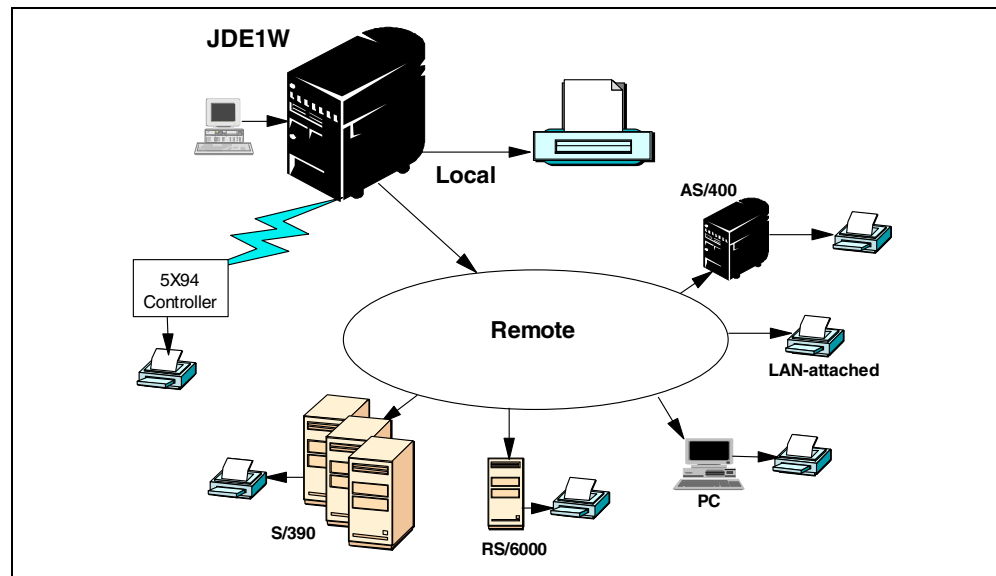


Figure 169. AS/400 printer attachment

14.1.1.1 Printers attached to WSC or 5x94

Several IBM printers (including SCS or IPDS) can be attached directly to AS/400 workstation controllers (WSC) by twinaxial cable. The same printers can also be attached by twinaxial cable to a Remote Control Unit IBM 5x94.

14.1.1.2 LAN attached IPDS printers

IPDS LAN capability is provided in a number of ways for IBM AS/400 printers. Any IPDS printer with an IBM Advanced Function Common Control Unit (AFCCU) can be LAN-attached to an AS/400 system (for example, IBM InfoPrint 60, InfoPrint 62, InfoPrint 3000, and InfoPrint 4000). These printers support one or more of the following attachments:

- TCP/IP Token-Ring
- TCP/IP Ethernet

- SNA Token-Ring
- SDLC

IBM workgroup printers such as Network Printer 12 (4312), Network Printer 17 (4317), Infoprint 20 (4320), or Infoprint 32/40 (4332) are LAN-attached with the appropriate network interface card (NIC), either TCP/IP Token-Ring, or TCP/IP Ethernet. The IPDS feature is also required.

The IBM 6400 line printer has an integrated Ethernet IPDS feature that provides Ethernet TCP/IP connection.

Older IBM IPDS printers (for example, IBM 3812, 3816, 3912, 3916, 3112, 3116, 4028, and 4230) can be LAN-attached using the IBM 7913 LAN Attachment unit (this is a hardware RPQ 8B4313 or 8B4314, depending on the printer). For details on LAN attachment of IPDS printers, refer to *the IBM Ethernet and Token Ring Configuration Guide*, G544-5240.

14.1.1.3 ASCII printers

The AS/400 system can use ASCII printers if they are attached to:

- A PC that is attached to the AS/400 system
- LAN

Printers attached to PCs

All ASCII printers can be connected to a PC using the standard parallel or serial cable. PC5250 printer emulation session is used to print AS/400 spooled files to a printer attached to a PC. When a spooled file is sent to a PC5250 printer session, it needs to be converted from EBCDIC to ASCII data stream supported by the PC printer. There are three ways that this conversion occurs:

- PC5250 transform based on a Printer Definition Table (PDT)
- PC5250 transform based on the Windows 95/Windows NT printer driver
- Host print transform

PC5250 is part of Client Access Express for Windows. For more information on Client Access Express, refer to Chapter 11, “Client Access Express and OneWorld” on page 283.

LAN attached printers

ASCII printers may be attached directly to the LAN using Token-Ring or Ethernet connections. Two system drivers (*IBMPJLDRV and *HPPJLDRV) allow you to attach ASCII printers on the LAN:

- The *IBMPJLDRV driver supports all of the IBM network printers. This includes the IBM Network Printer 12 (4312), Network Printer 17 (4317), Infoprint 20 (4320), and Infoprint 32/40 (4332) attached with a TCP/IP network interface card.
- The *HPPJLDRV driver supports all of the HP and HP compatible printers (for example, HP4 and HP5) attached using an HP JetDirect card (TCP/IP address).

14.1.2 AS/400 controllers and devices

The AS/400 system can automatically configure any controller or device (printer or display) attached to twinaxial or ASCII controllers. The same is done for devices attached to an IBM 5494-EXT remote controllers, attached using LAN or

WAN (Token-Ring or Ethernet). This is accomplished by selecting automatic configuration through the system value QAUTOCFG and using the Work with System Value (`WRKSYSVAL QAUTOCFG`) command.

Automatic configuration also occurs for any Client Access or Telnet devices. The system value that controls this is called QAUTOVRT. The `WRKSYSVAL` command allows the configuration change as in QAUTOCFG.

Automatic configuration for remote controllers can be achieved using the system value QAUTORMT. For additional information local device and automatic configuration, refer to *AS/400 Local Device Configuration*, SC41-5121. For additional information on remote workstation support, refer to *Remote Work Station Support*, SC41-5402.

14.1.3 AS/400 output queue

An output queue is a list of spooled files waiting to be printed. A single output queue may have spooled files from many different jobs and many different users. In some cases, a single job may place spooled files on more than one output queue. Spooled files from a single output queue can be directed to multiple printers.

14.1.4 Printer files

Whenever you print using the AS/400 spooling system, a printer file is used. A printer file determines how the system handles output from application programs. We can categorize printer files in two groups:

- **Program-described printer files:** These printer files do not have any field or record-level formatting. The attributes of the printer file are used to define how all of the data in the spooled file is printed. Any positioning of information within the file has to be determined by the application program. Most of the printer files use these simple files.
- **Externally-described printer files:** These printer files have formatting defined using Data Description Specification (DDS) external to the application program. Some of the attributes of the printer file apply to all of the data. The DDS can override or enhance these options for individual records or fields (for example, a single field can be printed as a barcode). For detailed information on DDS output keywords, refer to the *Data Description Specification Guide*, SC41-5712. An example of a system-supplied printer file is QSYSVRT in the QSYS library.

14.1.5 Spooled files

A spooled file contains the to-be-printed data. The spooled files stored in output queues can have different origins and different formats (data streams):

- Spooled files can be created on the AS/400 system by an application or by pressing the Print Screen key.
- Windows clients can send a PC output to an AS/400 output queue.
- You may also receive spooled files from host systems (IBM S/390), UNIX systems (IBM RS/6000), OEM systems, or other AS/400 systems.

14.1.5.1 Data stream types

The AS/400 system supports different data streams and can automatically create the majority of them. The Printer device type parameter in the printer file determines the type of data stream to be created (Figure 170).

```

                                Create Printer File (CRTPRTF)

Type choices, press Enter.

File . . . . . MYPRTF          Name
Library . . . . . *CURLIB      Name, *CURLIB
Source file . . . . . *NONE     Name, *NONE
Library . . . . .              Name, *LIBL, *CURLIB
Source member . . . . . *FILE   Name, *FILE
Generation severity level . . . 20
Flagging severity level . . . . 0
Device:
Printer . . . . . *JOB         Name, *JOB, *SYSVAL
Printer device type . . . . . *SCS      *SCS, *IPDS, *LINE...
Text 'description' . . . . . *SRCMBRTXT

                                Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel

```

Figure 170. Create Printer File: Printer device type parameter

The printer device type parameter can be set to one of the following values:

- ***SCS (SNA Character String):** SCS is used to control line mode printers and has a relatively simple structure.
- ***IPDS (Intelligent Printer Data Stream):** A server-to-printer data stream used for Advanced Function Printing (AFP) applications and reliable TCP/IP printing across the network. It provides an attachment-independent interface for managing IPDS printers. The IPDS connection provides an interactive, two-way dialog between the printer writer (PSF/400) and the printer that controls printer information, status, error recovery, and resource management. Note that the AS/400 generated IPDS is a subset of the full IPDS.
- ***AFPDS (Advanced Function Printing Data Stream):** A data stream for advanced function printers (independent of operating systems and page printers and portable across environments). AFPDS is a structured data stream divided into components called “objects”. AFPDS includes text, images, graphics, barcodes, and reference AFP resources (for example, overlays, page segments, and fonts).
- ***LINE (Line data stream):** A LINE data stream that uses a page definition and a form definition (standard AS/400 formatting objects) to format the spooled file.
- ***AFPDSLINE (AFPDS line data stream):** AFPDSLINE data stream is a mixture of AFP structured fields and LINE data. Only certain AFP structured fields can be mixed with the line data.

- ***USERASCII (ASCII data stream):** There is no formal structure that controls the use of the ASCII data stream to control printers attached to systems providing ASCII support. There is no architectural data stream standard to which ASCII printers can conform in the interest of uniformity.
A *USERASCII spooled file can contain any form of ASCII printer data stream (for example, PCL5, or PostScript).

The user should designate a printer that supports the data stream created in the spooled file.

By default, AS/400 printer files use the SNA character string (SCS) as the data stream type. This type of data stream can be sent to any printer, including ASCII printers using SCS-to-ASCII Host Print Transform. SCS spooled files can also be sent to printers configured as *IPDS, AFP=NO, and *IPDS, AFP=*YES. The printer writer handles this automatically. It looks at the printer's device description and transforms the SCS spooled file into the appropriate data stream. For IPDS printers configured AFP (*YES), the standard process includes the steps as illustrated in Figure 171.

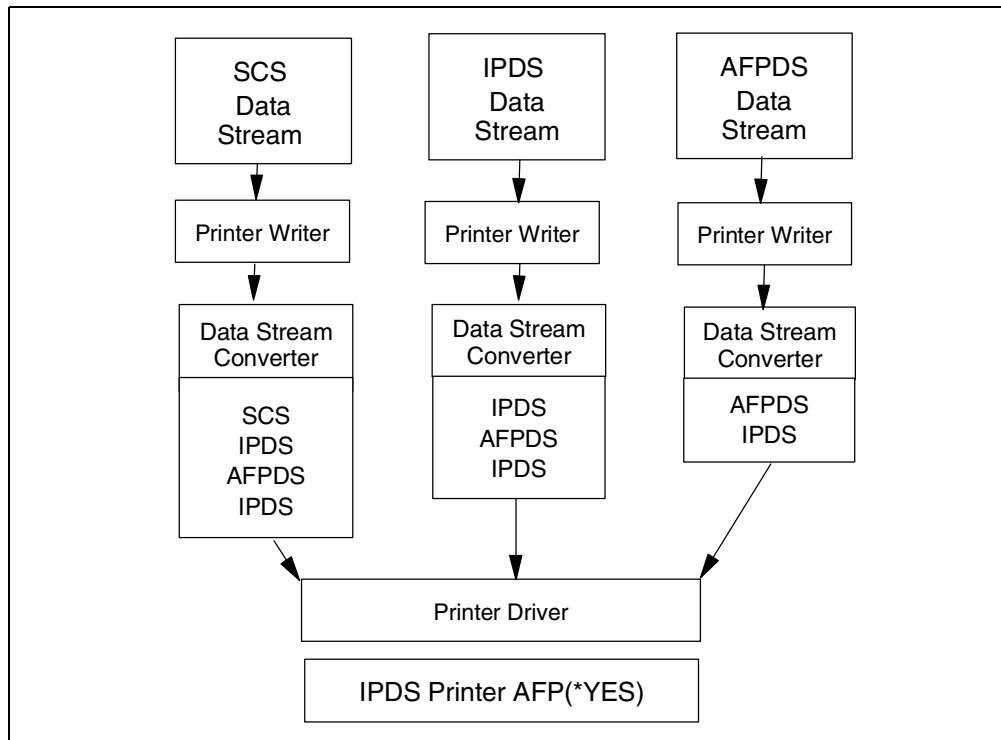


Figure 171. Data stream transforms when printing to an IPDS AFP (*YES) printer

14.1.5.2 AS/400 host print transform

The AS/400 host print transform function allows SCS-to-ASCII and AFPDS-to-ASCII conversion to take place on the AS/400 system instead of by 5250 emulators. Having the conversion take place on the AS/400 system provides the following advantages:

- **Consistent output for most ASCII printers:** The host print transform function is capable of supporting many different types of ASCII printer data streams (for example, the Hewlett-Packard Printer Control Language (PCL),

the IBM Personal Printer Data Stream (PPDS), and the Epson FX and LQ data streams). Having the conversion done on the AS/400 system ensures that the resulting ASCII printer data stream provides the same printed output regardless of the emulator or device to which the printer is physically attached.

- **Support for many different ASCII printers:** Currently, each emulator supports a limited number of ASCII printers. With the host print transform function, most IBM printers and a large number of OEM printers are supported.
- **Customized printer support:** Workstation customizing objects that come with the host print transform function can be updated by the user to change or add characteristics to a particular printer. Also, if the host print transform function does not have a workstation customizing object for a printer you want to use, you can create your own.

Figure 172 shows an overview of some of the ways in which ASCII printers can be attached. Host print transform can be used to print to all of these printers.

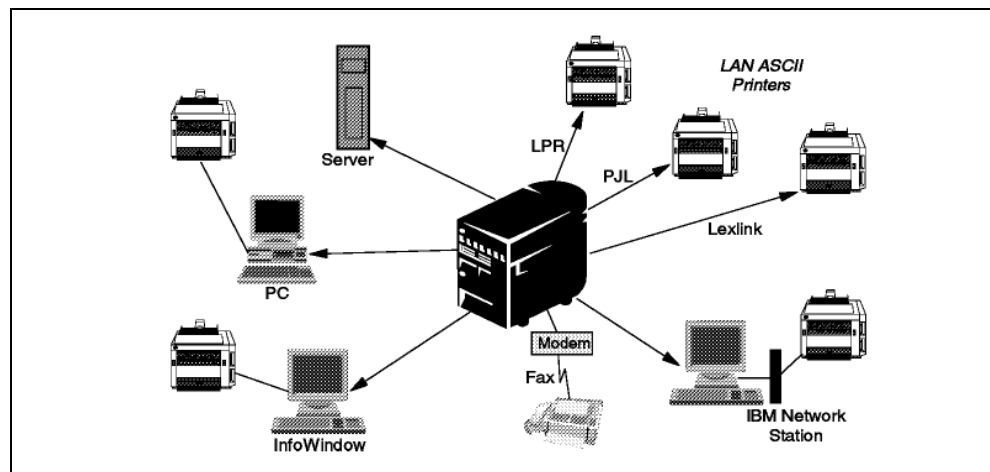


Figure 172. Host print transform

14.1.6 Printer writer

A printer writer is an OS/400 program that sends spooled files from an output queue to a printer defined on the AS/400 system. More than one printer writer can be started to the same output queue (10 is the limit). However, each writer name must be unique and of the same type.

The printer writer program takes spooled files, one at a time, from the output queue based on their priority. The printer writer program prints a spooled file only if its entry on the output queue indicates that it has a ready (RDY) status. You can display the status of a particular spooled file by using the Work with Output Queue (WRKOUTQ) command.

14.1.7 Remote system printing

Remote system printing is particularly useful for customers who have networked systems for automatically routing spooled files to printers connected to other systems. Output queue parameters define the target system. Depending on the target system or printer, the host print transform can be called to convert the spooled file into an ASCII printer data stream.

14.1.7.1 Remote writer program

A remote writer is an OS/400 program that takes spooled output files from a remote output queue and sends them to the specified remote system. The remote writer, which is a system job, sends the spooled output files using SNADS, TCP/IP, or Internetwork Packet Exchange (IPX). This function is known as remote system printing on the AS/400 system. The Start Remote Writer (STRRMTWTR) command is used to initiate remote system printing.

After the spooled output file is successfully sent to a remote system, it is deleted or saved as determined by the *SAVE* spooled file attribute value. Figure 173 shows several remote system printing possibilities.

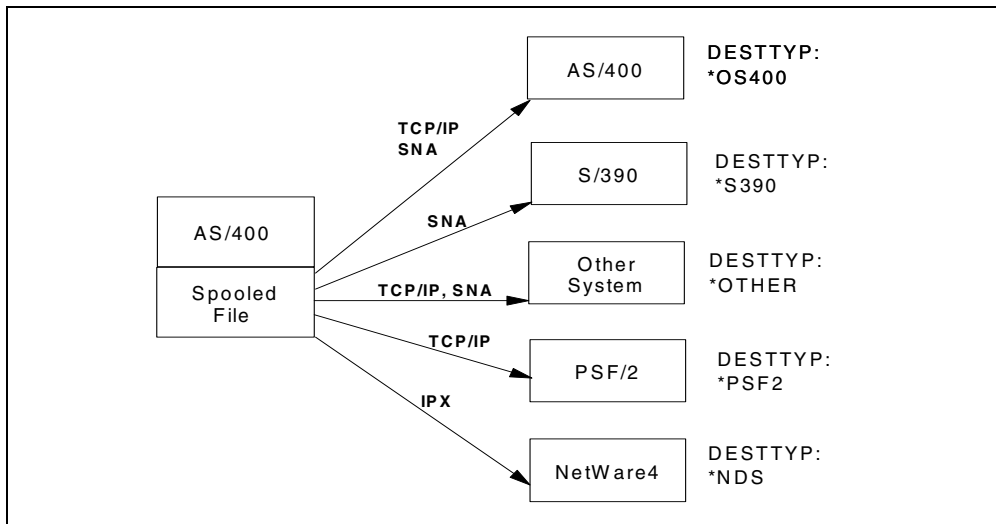


Figure 173. Remote system printing

14.1.7.2 TCP/IP LPR and LPD printing

You can request to have your spooled file sent and printed on any system in your TCP/IP network. The line printer requester (LPR) is the sending or client portion of a spooled file transfer. On the AS/400 system, the Send TCP/IP Spool File (SNDTCPSPLF) command, the TCP/IP LPR command, or remote system printing provide this function by allowing you to specify on which system you want the spooled file printed and how you want it printed. When sending a spooled file, the host print transform function can also be used to transform SCS or AFPDS spooled files into ASCII.

Printing the file is done by the printing facilities of the destination system. The destination system must be running TCP/IP. The line printer daemon (LPD) is the process on the destination system that receives the file sent by the LPR function. Figure 174 shows process flow of remote printing with LPR and LPD.

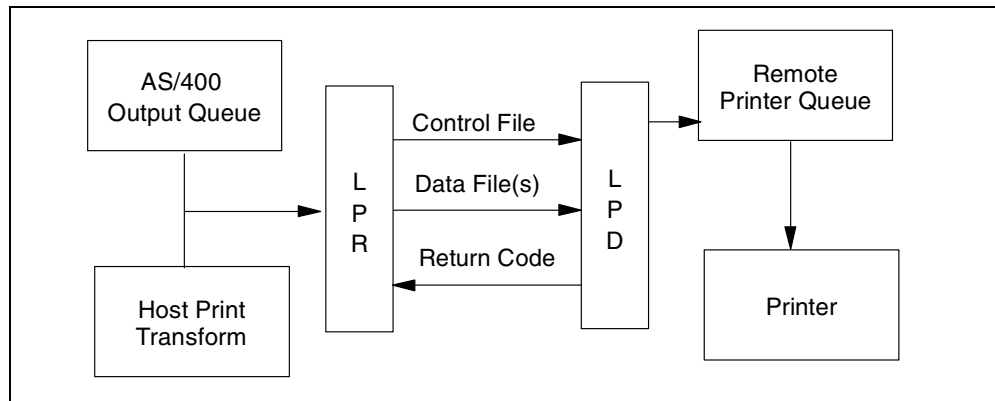


Figure 174. TCP/IP LPR and LPD

14.1.8 AS/400 NetServer

With AS/400 NetServer, you can make AS/400 output queues available as network printers to PC clients. AS/400 NetServer has been available since Version 4 Release 2 of OS/400. It is integrated into the base of OS/400 and is *not* a separately installable option or product.

What AS/400 NetServer is

NetServer enables an AS/400 system to provide file and print serving in a Windows network without the need to install additional hardware or software on the AS/400 system. AS/400 NetServer uses the Server Message Block (SMB) protocol to communicate with the network. PCs access the AS/400 NetServer with Windows networking functions without having additional software installed.

NetServer officially supports Windows for Workgroups (Windows 3.11), Windows 95, Windows 98, Windows NT 3.51, and Windows NT 4.0 clients. OS/2 Warp Connect, although it is an SMB client, is not supported with NetServer.

Sharing of printers

Sharing an output queue with NetServer works almost the same as sharing a directory. There are three methods to do this. The first two methods involve Operations Navigator, and the third method is a direct call to a AS/400 NetServer API on the AS/400 system.

For more information on NetServer, refer to *AS/400 Client Access Express for Windows: Implementing V4R4M0*, SG24-5191.

14.1.9 Client Access Express printing

In the Client Access Express environment, it is possible to print a PC application output on an AS/400 printer, and AS/400 application output on a PC printer. Or, by using a combination of these functions, you can print PC application output on another PC printer. For more information, refer to *Client Access Express for Windows – Setup V4R4M0*, SC41-5507, or *AS/400 Printing V*, SG24-2160.

When using Client Access Express for your printing needs, two different types of printing capabilities are provided:

- Sharing AS/400 printers using AS/400 NetServer
- Printer emulation

Windows drivers have to be used if the target printer is an ASCII printer. In this case, the spooled file in the AS/400 output queue is shown with a *USERASCII Device Type (DEVTYPE) attribute.

14.1.9.1 Sharing AS/400 printers using NetServer

Client Access Express for Windows, unlike the other Client Access clients, does not provide by itself the functionality of Network Drives and Network Printing. It relies on AS/400 NetServer for these functions. While Client Access for Windows 95/NT Network Printers use background jobs on the PC to make their functions available to the Windows desktop, NetServer does not need background jobs on the PC. This results in less overhead and less memory used on the PC.

For more information, refer to on NetServer, refer to 14.1.8, “AS/400 NetServer” on page 387.

14.1.9.2 Printer emulation

Printing AS/400 application output on PC attached printers using facilities such as Client Access Express is referred to as *printer emulation*. In this case, AS/400 spooled files in an SCS or an AFP data stream must be converted into an ASCII printer data stream depending on the target PC printer. This conversion can be done by one the following methods:

- **PC5250 emulation based on a Windows printer driver:** The transformation takes place on the PC, and only SCS spooled files can be converted. No customization is possible.
- **PC5250 emulation using Printer Definition Tables (PDT):** The transformation takes place on the PC, and only SCS spooled files can be converted. Printer functions can be adapted by modifying the Printer Definition Table (PDT). The modified PDT must be the Printer Definition Table (PDT). The modified PDT must be available on all PCs using the printer.
- **OS/400 host print transform:** The transformation takes place on the AS/400 system. SCS and AFPDS spooled files can be converted. Customization is possible by modifying the Work Station Customizing (WSCST) object. The same WSCST object is used for all printers of a similar type. For detailed information on host print transform, see 14.1.5.2, “AS/400 host print transform” on page 384.

Redirecting PC application output using the AS/400 system to another PC printer in the network is a combination of the previous two capabilities. PC generated output is sent to an AS/400 output queue in an ASCII printer data stream and then printed on a Client Access attached ASCII printer. This brings the AS/400 spooling capabilities to PC application output.

14.1.10 Spool management

This section describes some actions used to manage printed output directed to the AS/400 spooling system. It includes:

- Creating remote output queues for printing to LAN attached printer
- Creating virtual printer devices
- Starting printer writers
- Working with output queues

14.1.10.1 Creating *remote* output queues

The following section describes the steps to create AS/400 remote output queues to support server printing. The section does not discuss how to set up a local Twinax attached printer in this book. Refer to Section 3.4, “Configuring Local Twinaxial Printers” in *AS/400 Local Device Configuration*, SC41-5121, for assistance.

You can create an AS/400 output queue for a LAN attached network printer after your initial software installation. To do so, select option **7** from the A98OWMNU - OneWorld Installation menu or type (Figure 175):

GO JDEOW/A98OWMNU

Note

If you are not signed on as user ONEWORLD, you need to use the `ADDLIBLE LIB(JDEOW)` command prior to going to the A98OWMNU menu.

```
A98OWMNU                J.D. Edwards & Company                ASM23
  B733                  OneWorld Installation Menu

... DATA LIBRARIES
  2.  Install Data Dictionary

... SYSTEM LIBRARIES
  4.  Install OneWorld Server Libraries
  5.  Install Additional Path Codes

... PRINTER CONFIGURATION
  7.  Create Output Queue

... POST INSTALLATION
  9.  Edit INI File
 10.  Verify Library List

... SERVER CONFIGURATION
 12.  Porttest
 13.  Start JDE Server
 14.  End JDE Server
 15.  Clear IPC

... MISC. OPERATIONS
 17.  Display Active Jobs
 18.  Work with Spooled Files

... MISC. ONEWORLD UTILITIES
 20.  Remove OneWorld Libraries
 21.  Save/Restore OneWorld Lib.

Selection or command
====> 7

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
F13=Information Assistant  F16=AS/400 main menu
```

Figure 175. Create output queue (Part 1 of 2)

On the next display that appears (Figure 176 on page 390), enter the following parameter values:

- **Outqueue Name:** The name for your new output queue. Substitute your printer name instead of JDEPRINTER. It should be something meaningful. For example, it can be the model and type of the printer (HP5SI), the physical location (FLR3RM202), or department (ACCOUNTING).
- **Printer IP Address:** The IP (Internet Protocol) address of the network card in the printer. If you do not know this information, check the configuration on the printer using the manual supplied with it or ask the local network administrator.
- **Remote Printer Queue:** Specify `PASS`. If you experience formatting problems, try using `TEXT`, or consult your printer manual for additional information.

This creates a remote output queue in library QGPL for you. You can also go directly to this display by typing `GO JDEOW/A98OWMNU` and pressing the Enter key.

```
                Create OneWorld Outqueue (C98OWOUT)

Type choices, press Enter.

Outqueue Name . . . . . JDEPRINTER      Character value
Printer IP Address . . . . . 9.99.99.999
Remote Printer Queue . . . . . 'PASS'

                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figure 176. Create output queue (Part 2 of 2)

If your printed output comes out incorrectly, try changing the Host Print Transform parameter in the Output Queue to `*YES`. To do so, enter the command:

```
CHGOUTQ QGPL/JDEPRINTER TRANSFORM(*YES)
```

You may want to auto start the print writer associated with the output queue and add a text description by typing:

```
CHGOUTQ OUTQ(QGPL/JDEPRINTER) AUTOSTRWTR(1) TEXT('J D Edwards OneWorld
Printer')
```

Press Enter.

This remote writer runs in the AS/400 subsystem called QSPL. When a spooled file is in a ready state and the remote writer is started or active, the AS/400 system sends the output to the LAN attached printer with this IP address.

Once a print writer is started, you must end it before you can make changes to the output queue. Type:

```
ENDWTR JDEPRINTER *IMMED
```

Press Enter.

After you complete making any changes, the writer should start up automatically. If not, you can manually start it by entering:

```
STRMTWTR QGPL/JDEPRINTER
```

Output queues can have a physical printer associated with them, or you can create an output queue to be used just as a report repository. The printouts or

spooled files reside there until printed, deleted, or moved to another output queue.

With special authority granted to this output queue, other LAN or WAN users can share this printer from their desktop.

For additional information on output queues, refer to Chapter 5, “Spool Support” in *Data Management*, SC41-5710.

14.1.10.2 Creating a virtual IPDS printer device

When printing using remote writer support, you do not have any direct communication with the printer itself. The spooled file is simply being sent to an IP address. Problem determination and error recovery is limited. There is no feedback on printer status, no management of print resources (for example, fonts, overlays and images), and an incomplete implementation of AS/400 printer file options.

Using an IPDS LAN-attached printer and defining an IPDS printer device description addresses all of these shortcomings of remote writer support. An IPDS printer device description provides, in essence, the functionality of a direct-attached (read Twinax) AS/400 printer in a network-connected environment.

You can create a printer device description by typing `CRITDEVPRT` and prompt it by pressing the F4 key. The display shown in Figure 177 appears.

```

                                Create Device Desc (Printer) (CRITDEVPRT)

Type choices, press Enter.

Device description . . . . . > JDEPRINT  Name
Device class . . . . . > *LAN           *LCL, *RMT, *VRT, *SNPT, *LAN
Device type . . . . . > *IPDS          3287, 3812, 4019, 4201...
Device model . . . . . > 0             0, 1, 2, 3, 4, 10, 13, 301...
LAN attachment . . . . . *IP           *LEXLINK, *IP, *USRDFN
Advanced function printing . . . *YES   *NO, *YES
Port number . . . . . > 5001          0-65535
Online at IPL . . . . . *YES          *YES, *NO
Font:
  Identifier . . . . . > 11           3, 5, 11, 12, 13, 18, 19...
  Point size . . . . . *NONE          000.1-999.9, *NONE
Form feed . . . . . *TYPE            *TYPE, *CONT, *CUT, *AUTOCUT
Separator drawer . . . . . *FILE      1-255, *FILE
Separator program . . . . . *NONE     Name, *NONE
Library . . . . . *NONE              Name, *LIBL, *CURLIB
Printer error message . . . . . *INQ  *INQ, *INFO

More...
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 177. The Create Device Description display (Part 1 of 3)

The parameters with the “>” mark show which ones were changed. Leave the defaults in the other fields. You have to press the Enter key to expand the hidden additional parameters after you fill out the fields on the display. See Figure 178 and Figure 179 on page 392.

```

Create Device Desc (Printer) (CRTDEVPRT)

Type choices, press Enter.

Message queue . . . . . QSYSOPR      Name, QSYSOPR
  Library . . . . . *LIBL          Name, *LIBL, *CURLIB
Activation timer . . . . . 170        1-2550, *NOMAX
Image configuration . . . . . *NONE      *NONE, *IMGA01, *IMGA02...
Maximum pending requests . . . . . 6      1-31
Print while converting . . . . . *YES      *NO, *YES
Print request timer . . . . . *NOMAX    1-3600, *NOMAX
Form definition . . . . . F1C10110    Name
  Library . . . . . *LIBL          Name, *LIBL, *CURLIB
Remote location:
  Name or address . . . . . > '9.99.99.999'

User-defined options . . . . . *NONE      Character value, *NONE
  + for more values

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 178. Create Device Description (Printer) display (Part 2 of 3)

```

Create Device Desc (Printer) (CRTDEVPRT)

Type choices, press Enter.

User-defined object:
  Object . . . . . *NONE      Name, *NONE
  Library . . . . .          Name, *LIBL, *CURLIB
  Object type . . . . .          *DTAARA, *DTAQ, *FILE...
Data transform program . . . . . *NONE      Name, *NONE
  Library . . . . .          Name, *LIBL, *CURLIB
User-defined driver program . . . *NONE      Name, *NONE
  Library . . . . .          Name, *LIBL, *CURLIB
Text 'description' . . . . . > 'J D Edwards Network Printer'

Additional Parameters

Remote network identifier . . . *NETATR      Name, *NETATR, *NONE
Workstation customizing object *NONE      Name, *NONE
  Library . . . . .          Name, *LIBL, *CURLIB

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 179. Create Device Description (Printer) display (Part 3 of 3)

14.1.10.3 Starting AS/400 printer writers

Before a printer can print the spooled files, a printer writer program must be started using either the autostart writer option or one of the following commands:

- STRPRTWTR: Start Printer Writer
- STRRMWTR: Start Remote Writer

Normally, all writers are started when the system starts after an Initial Program Load (IPL). This is done using the system value QSTRPRTWTR (Start printer writers at IPL). The shipped value of QSTRPRTWTR is 1, which starts all writers. A change to this value becomes effective only at the next IPL. The writer commands can be accessed using the Command Writer (GO CMDWTR) command.

If a writer ends the request of a user or operator, it must be started before it can print, unless requesting direct printing and bypassing the output queue. To end a print writer (either a local or a remote), on the command line, enter:

```
ENDWTR JDEPRINT *IMMED
```

Local printer writer

Figure 180 and Figure 181 on page 394 show how to start a printer writer for a locally attached printer.

```

                                Start Printer Writer (STRPRTWTR)

Type choices, press Enter.

Printer . . . . . > PRT01      Name, *ALL, *SYSVAL
Output queue . . . . . > JDEOUTQ Name, *DEV
  Library . . . . .           Name, *LIBL, *CURLIB
Queue for writer messages . . . > *REQUESTER Name, *DEV, *REQUESTER
  Library . . . . .           Name, *LIBL, *CURLIB
Form type options:
  Form type . . . . . *ALL      Form type, *ALL, *STD, *FORMS
  Message option . . . . . *INQMSG *INQMSG, *MSG, *NOMSG...
File separators . . . . . *FILE  0-9, *FILE
Drawer for separators . . . . . *DEV  1-255, *DEV, *FILE

                                Additional Parameters

Writer . . . . . *DEV          Name, *DEV

                                                                More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 180. STRPRTWTR prompt command (Part 1 of 2)

```

                                Start Printer Writer (STRPRTWTR)

Type choices, press Enter.

Auto-end options:
  Automatically end writer . . . *NO          *NO, *YES
  If yes, when to end . . . . . *NORDYF     *NORDYF, *FILEEND
  Allow direct print . . . . . *NO          *NO, *YES
  Align page . . . . . *WIR              *FILE, *WIR, *FIRST
  Initialize printer . . . . . *WIR              *WIR, *FIRST, *ALL
  Spooled file . . . . . *NONE           Name, *NONE, *LAST
  Job name . . . . . *                   Name, *
  User . . . . .                               Name
  Number . . . . .                               000000-999999
  Spooled file number . . . . . *ONLY       1-9999, *ONLY, *LAST
  Starting page . . . . . *BEGIN          Number, *BEGIN

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 181. Additional information for the STRPRTWTR command (Part 2 of 2)

Most of the parameters needed for a user or operator to start a printer writer are on the first display (Figure 180 on page 393). The information needed includes the name of the writer, the name of the output queue (can be the writer name or a different name), and where the printer attention messages are sent (for our example, the user or operator that started the writer). There are other parameters that control what form types to print, separator pages to be printed, and a name for the writer (if the device name is not used).

The second display (Figure 181) specifies if the printer should end when no ready files are available, allow direct print (used to allow a program to direct print to the device bypassing the queue; this can be used in a barcode scenario), check page alignment, and initialize the writer. The other parameters include:

- Spooled file
- Job name
- User
- Number
- Spooled file number
- Starting page

These parameters are used to print or reprint a specific spooled file. The information for these parameters are retrieved by using the Work Output Queue (WRKOUTQ) command and finding the spooled file to print or reprint. The starting page parameter is used to start at a specific page in the spooled file for print or reprinting.

Remote printer writer

Figure 182 shows how to start a printer writer for a remotely attached printer.

```

Start Remote Writer (STRRMIWTR)

Type choices, press Enter.

Output queue . . . . . > JDEPRINTER Name, *ALL
  Library . . . . . *LIBL Name, *LIBL, *CURLIB
Queue for writer messages . . . *requester Name, *OUTQ, *REQUESTER
  Library . . . . . Name, *LIBL, *CURLIB
Form type options:
  Form type . . . . . *ALL Norm type, *ALL, *STD, *FORMS
  Message option . . . . . *NOMSG *NOMSG, *INQMSG, *MSG...

Additional Parameters

Writer . . . . . *OUTQ Name, *OUTQ, *SYSGEN
Auto-end options:
  Automatically end writer . . . *NO *NO, *YES
  If yes, when to end . . . . *NORDYF *NORDYF, *FILEEND

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figure 182. Start Remote Writer (STRRMTWTR)

The STRRMTWTR command is different. It starts the writer using an output queue name that was created as a remote queue. Some of the parameters are the same as the STRPRTWTR command. The command specifies the queue for messages, form type, writer name, and auto-end options.

Once the remote writer is started, all files in a ready status are sent or printed to the remote printer or remote system.

14.1.10.4 Work with output queues

After the queue is created, you need to know how to display or view the contents using the Work with Output Queues (WRKOUTQ) command. The WRKOUTQ command shows the contents of a specific queue by indicating the queue name, all output queues by specifying the special value *ALL, or all output queues in a specific library by specifying the special value *ALL and the library name. See Figure 183 on page 396.

```

Work with All Output Queues

Type options, press Enter.
 2=Change  3=Hold  4=Delete  5=Work with  6=Release  8=Description
 9=Work with Writers  14=Clear

Opt  Queue      Library      Files  Writer      Status
MRK  MRK          MRK          0      JDEPRINTER  RLS
QDMT QDMT          QGPL         0      JDEPRINTER  RLS
JDEPRINTER QGPL         0      JDEPRINTER  RLS
QDKT QGPL         0      JDEPRINTER  RLS
QPRINT QGPL         6      JDEPRINTER  RLS
QPRINTS QGPL         0      JDEPRINTER  RLS
QPRINT2 QGPL         0      JDEPRINTER  RLS
JDEPRINT QUSRSYS      0      JDEPRINT    RLS
ONEWORLD QUSRSYS      1      JDEPRINT    RLS
QEZDEBUG QUSRSYS      0      JDEPRINT    RLS
QEZJOBLOG QUSRSYS      69     JDEPRINT    RLS

Command
====>
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F24=More keys
Bottom

```

Figure 183. Example of the Work with All Output Queues display

From the display shown in Figure 183, you can access all output queues. Through the options shown in Figure 183, you can change output queue attributes; hold an entire output queue; delete, work with (and display stored spooled files), and release the queue; change the description of the queue; or clear all spooled files stored in the queue.

```

Work with Output Queue

Queue:  ONEWORLD      Library:  QUSRSYS      Status:  RLS

Type options, press Enter.
 1=Send  2=Change  3=Hold  4=Delete  5=Display  6=Release  7=Messages
 8=Attributes  9=Work with printing status

Opt  File      User      User Data  Sts  Pages  Copies  Form Type  Pty
QSYSPRT  QSECOFR  QSECOFR  RDY  1  1  *STD  5
QSYSPRT  QSECOFR  QSECOFR  RDY  1  1  *STD  5
QSYSPRT  QSECOFR  QSECOFR  RDY  1  1  *STD  5
QSYSPRT  QSECOFR  QSECOFR  RDY  1  1  *STD  5
QPJOBLOG QSECOFR  QPADEV001  RDY  2  1  *STD  5
QSYSPRT  QSECOFR  WRKSYSVAL  RDY  3  1  *STD  5
QSYSPRT  QSECOFR  WRKSYSVAL  RDY  2  1  *STD  5

Parameters for options 1, 2, 3 or command
====>
F3=Exit  F11=View 2  F12=Cancel  F20=Writers  F22=Printers
F24=More keys
Bottom

```

Figure 184. Example of option 5 on the ONEWORLD output queue

Figure 184 shows the Work with Output Queue display. This display allows you to work with individual or groups of spooled files. In the top right-hand corner, the

Status field refers to the status of the output queue (RLS - Released) and the status of the printer writer (WTR - Writing) in this example. The following list shows all of the output queue status possibilities:

- **HLD (Held):** The queue is held.
- **HLD/WTR (Held/Writer):** The queue is attached to a writer and is held.
- **RLS/WTR (Release/Writer):** The queue is attached to a writer and is released.
- **RLS (Released):** The queue is released, and no writer is attached.

14.1.10.5 AS/400 Printer menu

The AS/400 Printer menu (Figure 185) can be helpful to a user who is not familiar with the system. It can be accessed by entering `GO PRINTER` on a command line. This menu has options to work with spooled files, output queues, printers, and writers. Remember, a user always has access to spooled files that they created. However, if you want to manipulate other printing features, you need *SPLCTL special authority.

```
PRINTER                                Printer                                System: ASM23
Select one of the following:
    1. Work with spooled output files
    2. Work with printers
    3. Change job printer
    4. Work with printer device status
    5. Work with output queues
    6. Display default system printer
    7. Change default system printer
    70. Related commands
Selection or command
===>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=Information Assistant
F16=AS/400 Main menu
(C) COPYRIGHT IBM CORP. 1980, 1998.
```

Figure 185. AS/400 Printer menu

14.2 OneWorld printing

This section describes the basic setup of a printer in the OneWorld environment.

OneWorld AS/400 servers generate PostScript reports in EBCDIC spooled files. The OneWorld code converts the PostScript report from EBCDIC to ASCII PDF format (Adobe's Portable Document Format). This format is then transformed via print filters into three additional ASCII formats: PCL, PostScript, or Line depending on the type of printer that prints the report.

After the report is created as a PDF file, it can be viewed from a workstation or printed. You view and print a report using Adobe Acrobat Reader.

In next release of OneWorld, the support for Advanced Function Presentation (AFP) will be added. AFP is a system and printer independent data stream designed for business output, volumes, and network environments. At the time this edition was published, the AFP data stream is converted into a very similar, but printer-specific Intelligent Printer Data Stream (IPDS). IPDS printing is completely system-managed, meaning that the server actively manages resources, printer status, and full error recovery (to the page level). IPDS facilitates this by being an interactive print data stream. The 2-way dialog with the server ensures a reliable printing environment even across or within TCP/IP networks. The system management is handled by either Print Services Facility (PSF/400 for the AS/400) or Infoprint Manager, depending on the server.

14.2.1 Creating a OneWorld printer

To get to the Printers menu, complete the following steps:

1. From the main OneWorld display, type GH9013 in the Fast Path box in the upper right-hand corner.
2. Press the Enter key. The Printers menu (GH9013) is a single point of entry for configuring your printers within OneWorld. Your display should appear as shown in Figure 186.

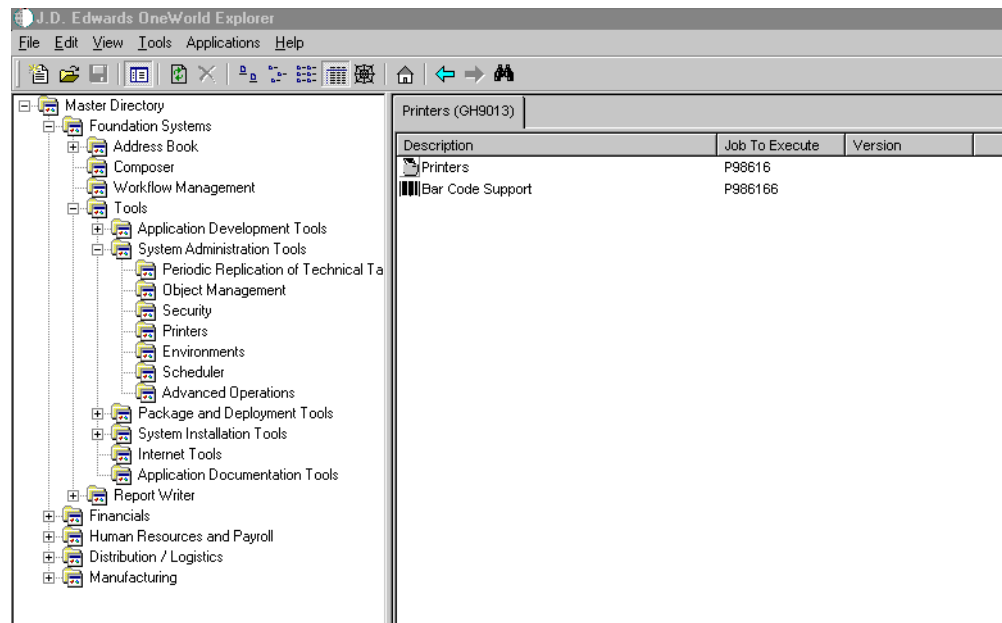


Figure 186. J.D. Edwards OneWorld Explorer printer menu

3. From the Printers menu, double-click the **Work With Physical Printers** option to select it (Figure 187).

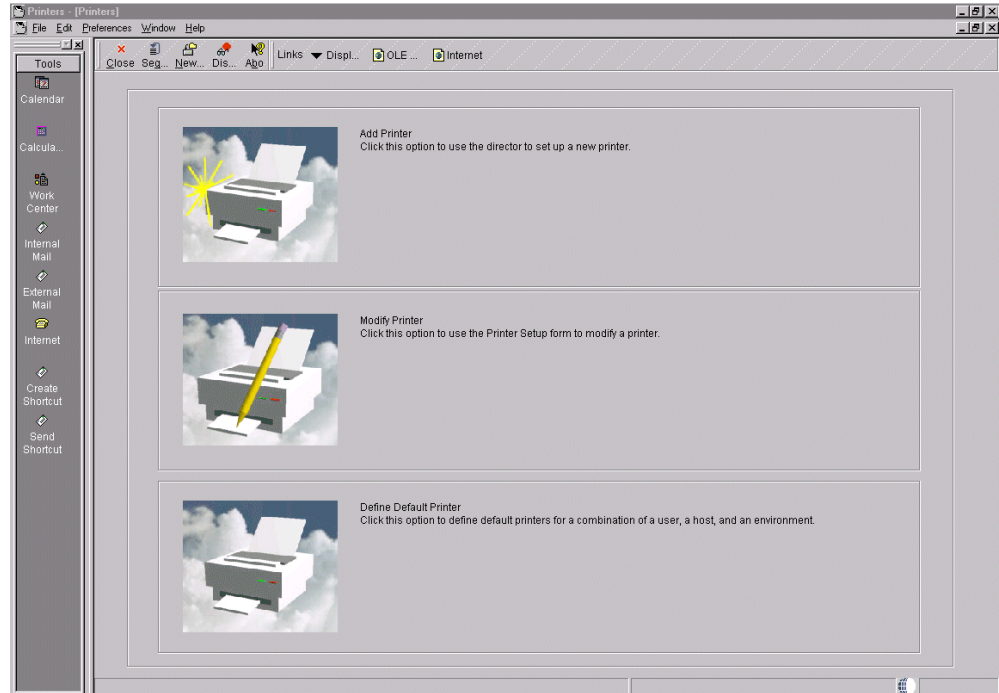


Figure 187. Work with physical printers menu (Part 1 of 2)

- From the Printers Window, click the **Add Printer** button (the printer picture with a star on the left side (Figure 188)).

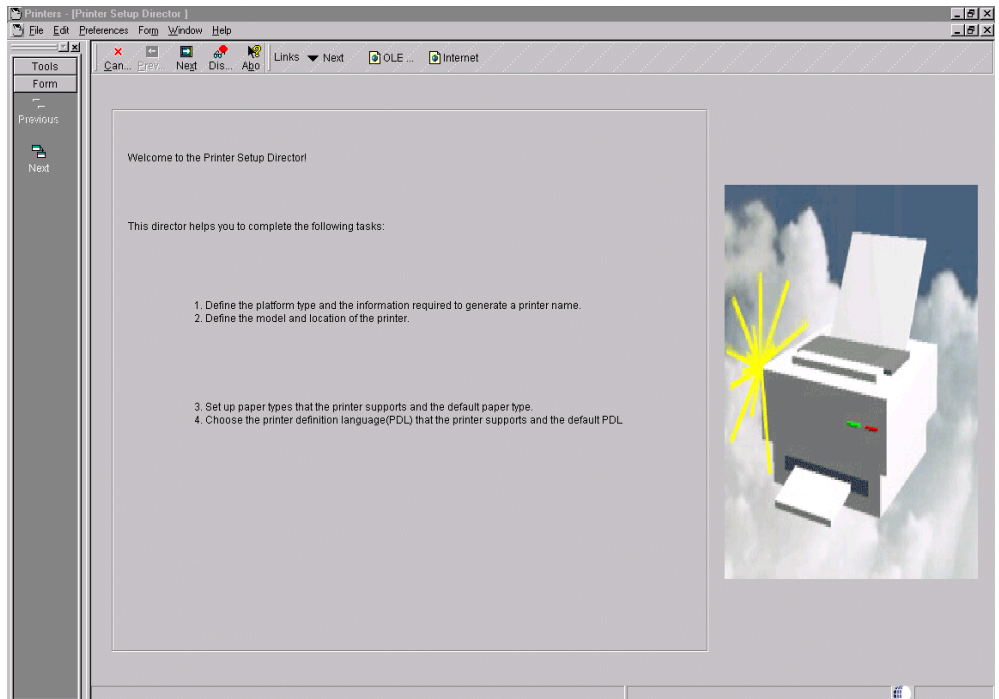


Figure 188. Work with physical printers menu (Part 2 of 2)

- On the following display, click on the **Next** button on the tool bar (Figure 189 on page 400).

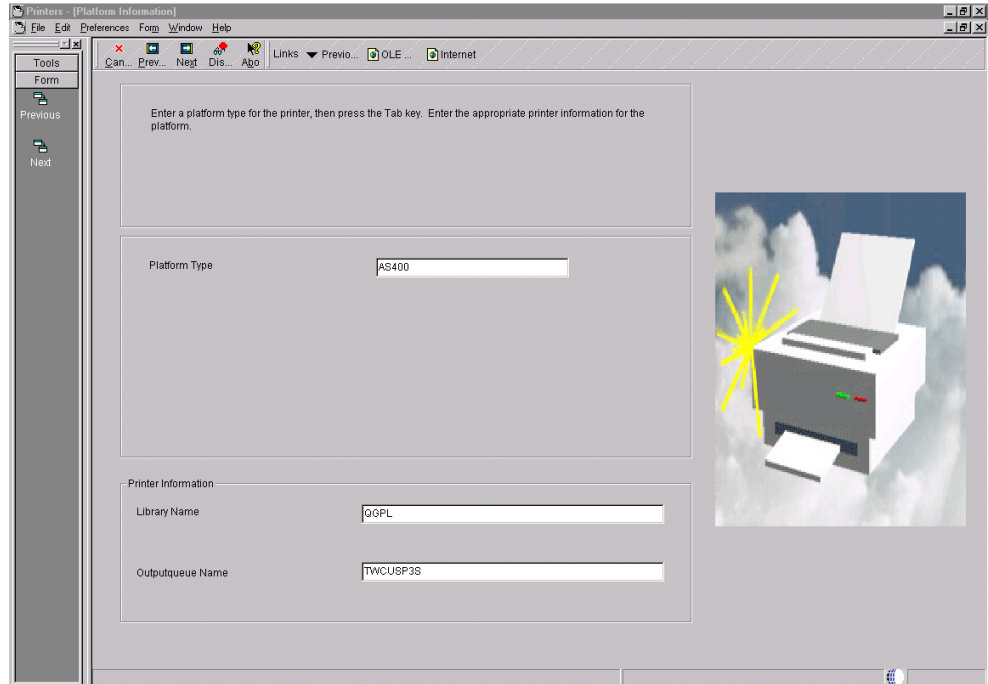


Figure 189. Printers (Platform Information) display (Part 1 of 3)

6. Enter the proper Platform Type, Library Name where the output queue resides, and the output queue Name. Then, click the **Next** button (Figure 190).

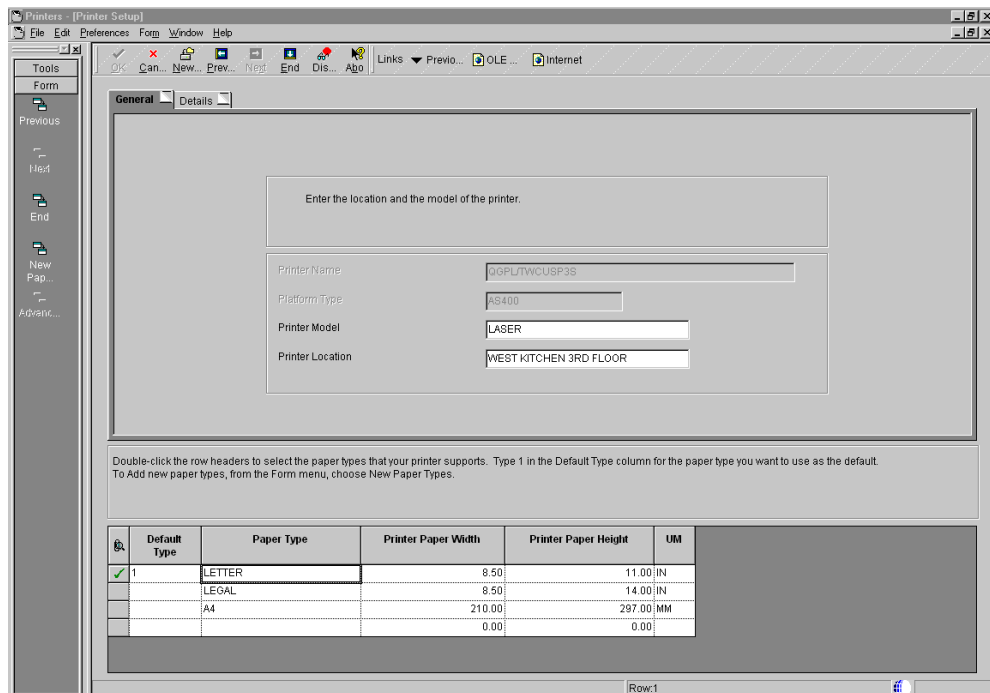


Figure 190. Printers (Platform Information) display (Part 2 of 3)

7. From here, enter the Printer Model (Laser is the only option available in B73.3.2) and Printer Location (text description field). Then, enter 1 in the Default Type column for the paper type. Double-click the gray box to the left of

that grid row so a green check mark shows appears on that item. Click the **Details** tab (Figure 191).

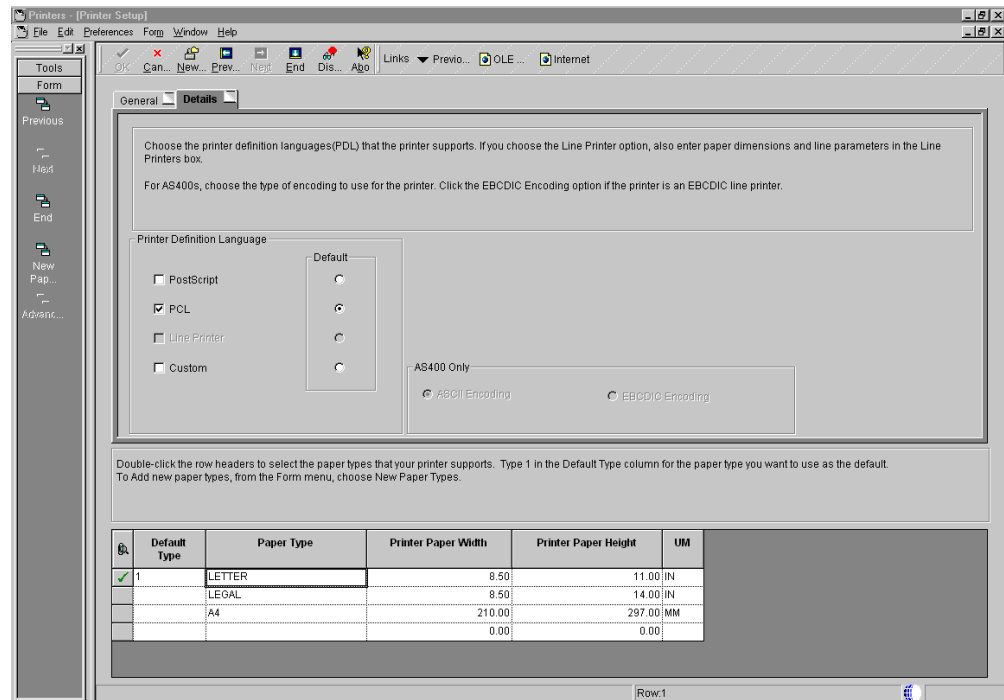


Figure 191. Printers (Platform Information) display (Part 3 of 3)

8. Choose the proper Printer Definition Language for your printer (if you choose Line Printer, the proper parameters must be set), if more than one definition is selected make sure you properly indicate the appropriate Default. If you want to verify the proper conversion string, choose **Custom**. Then, from the Form menu, choose **Advanced**. To know what the appropriate conversions strings are, reference the *OneWorld System Administration Guide* for B73.3.2. Then, click the **End** button on the tool bar. This takes you back to the display shown in Figure 187 on page 399.
9. You should be back to the main printer setup window now, and you can now set the just defined printer to be a default printer for the appropriate user/group. Click the **Define Default Printer** button (the printer picture that is just the printer). See Figure 192 on page 402.

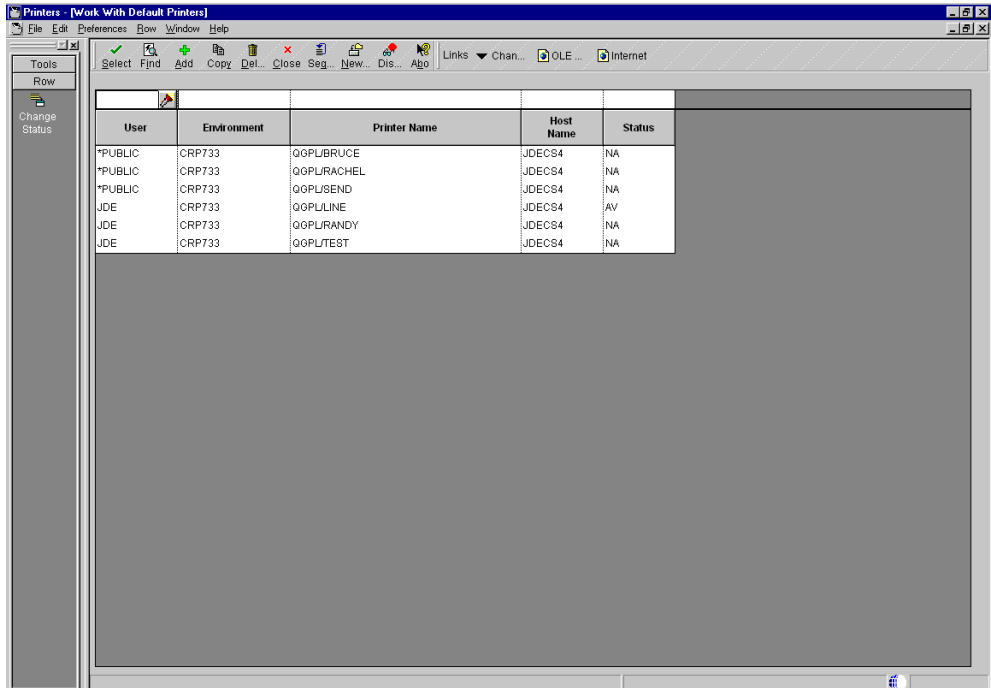


Figure 192. Wok with Default Printers

10. Click the **Add** button on the tool bar (Figure 193).

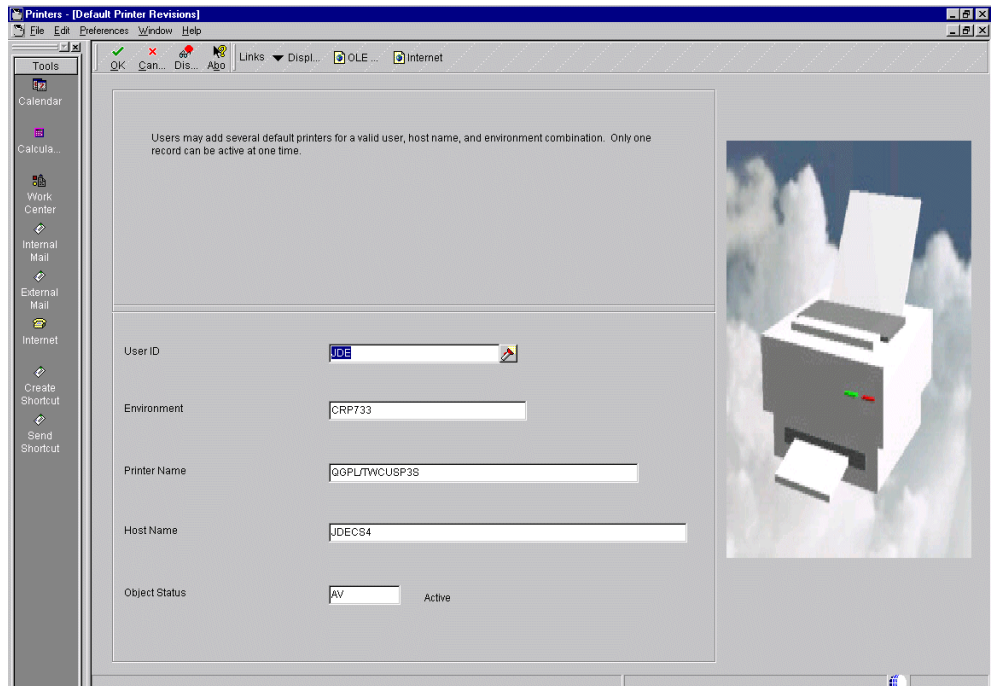


Figure 193. Default Printer

11. Enter the appropriate data into the User ID, Environment, Printer Name (library/output queue), Host Name (Enterprise Server), and Object Status (Only one default printer can be active for a given user). Click the **OK** button.

12. This takes you back to the display shown in Figure 192. Click **Find** to see your newly added default printer.

14.2.2 Testing printing in OneWorld

To test the printing capabilities in OneWorld, follow this process:

1. From the OneWorld main menu, type either P98305 or BV in the Fast Path field in the upper right-hand corner of the display. Press Enter.
2. At the Work with Batch Versions display, you can click the flashlight icon to show a listing of the available reports. In this example, we use the R0010P - Company Constants Report. This is a fast running report that does not need additional processing options.
3. Click the **Find** button, and select Version **ZJDE0001**. Your display should appear as shown in Figure 194.

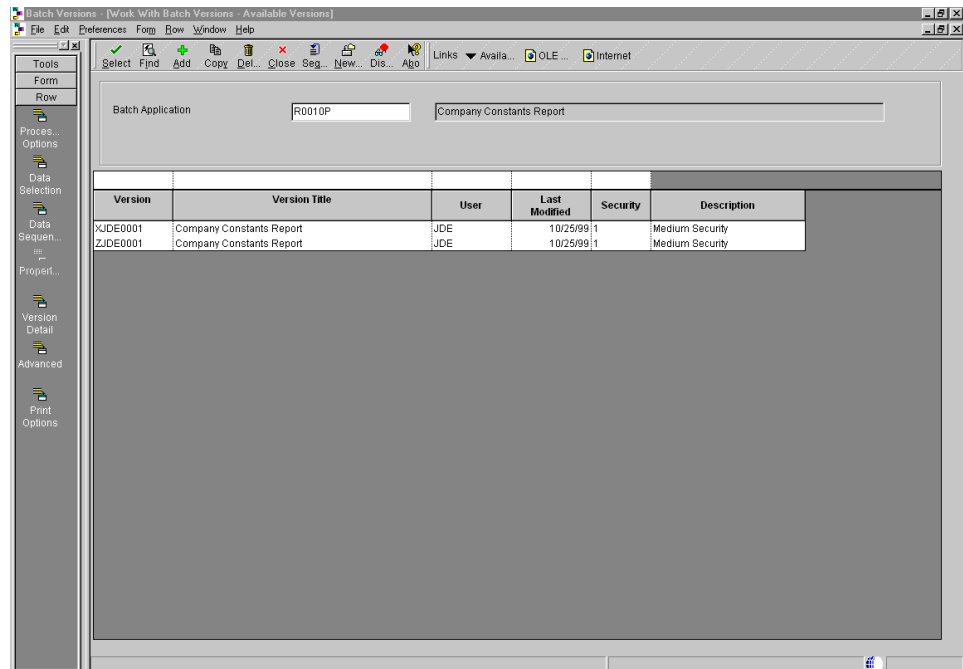


Figure 194. The Work with Batch Versions display

If you do not see the toolbar on the left-hand side of your display, click **Preferences** on the top of your screen, and select **Exit Bar**. Click the **Select** button. You should now see the Version Prompting display.

4. From this display, you can prompt for additional Data Selection or Sequencing options. You can also override your default location or turn error logging on by clicking the **Advanced** icon in the left (Row) toolbar. For our example, we accept all of the defaults by clicking the Submit button. The next display appears as shown in Figure 195 on page 404.

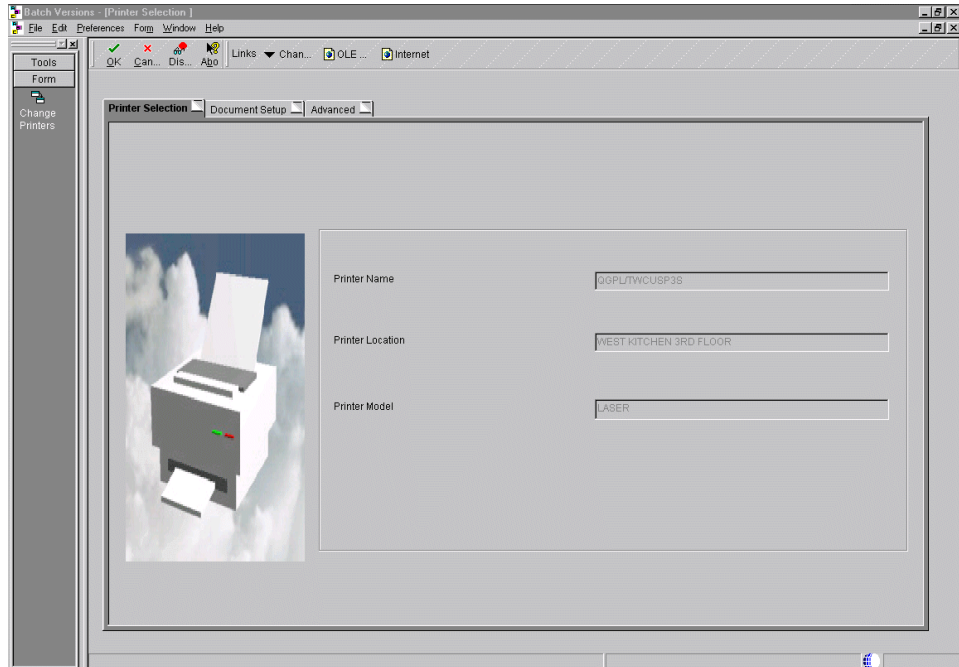


Figure 195. Printer Selection display

5. In the Printer Selection window, the Printer Selection Tab displays the default printer for your user ID. You have the option of changing the printer to which your output will go by selecting change printers from the Form menu (Figure 196).

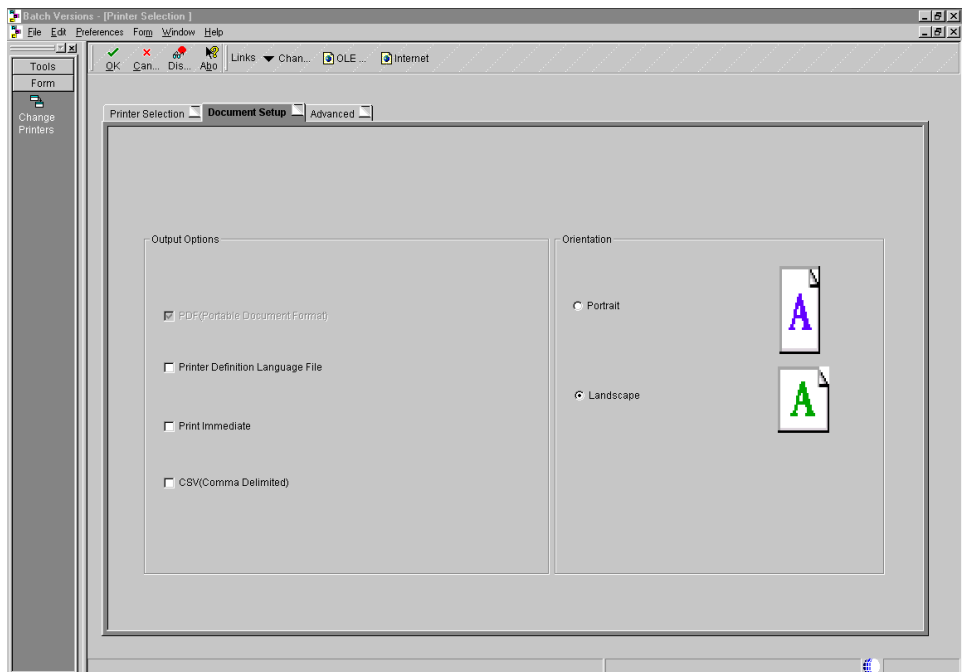


Figure 196. Printer Selection

6. You also have the opportunity to change the report output by selecting the Document Setup Tab in the Printer Selection Window (Figure 197).

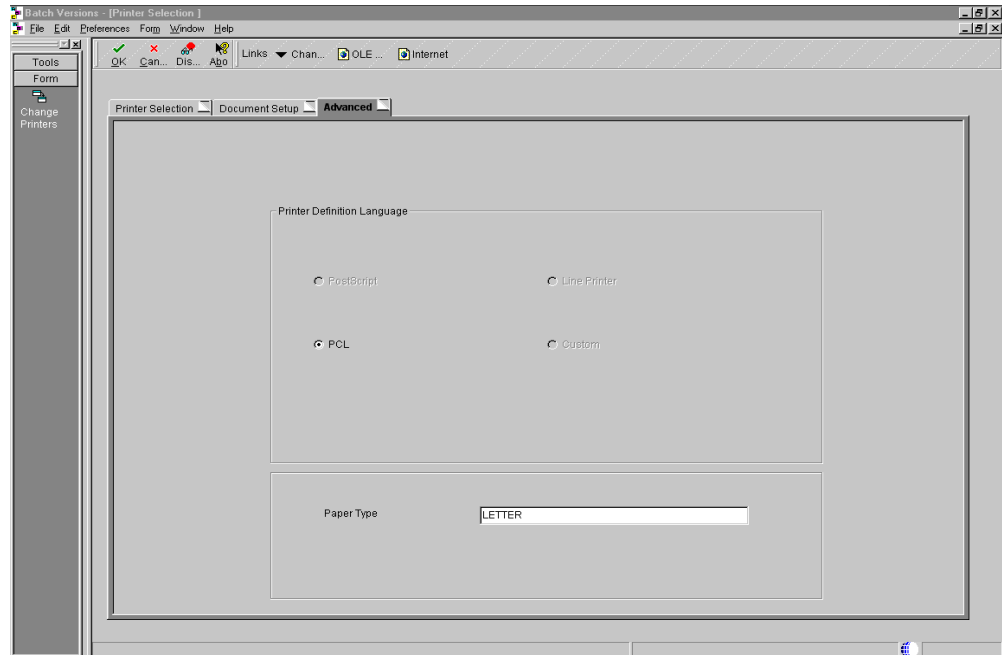


Figure 197. Printer Selection display

7. There are some additional parameters that may be changed in the Advanced Tab in the Printer Selection window.
8. Click **OK** to run the report. The batch job should now be submitted to job queue defined by in your OneWorld workstation jde.ini file, for example, QB7332, on the AS/400 system.
9. If you want to monitor the job on the system, type the following command and press Enter:

```
WRKACTJOB SBS (QBATCH)
```

You should see a display like the one shown in Figure 198 on page 406.

```

Work with Active Jobs
JDECS4
03/29/00 19:49:52
CPU %: 2.4 Elapsed time: 00:01:40 Active jobs: 134

Type options, press Enter.
2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
OBATCH QSYS SBS .0 DEQW
R0010P QNOTES BCI .0 RUN

Parameters or command
====>
More...

```

Figure 198. Work with Active Jobs display

10. From the OneWorld main menu, Fast Path to the Work With Servers menu (P986116).

11. Click the **Find** button to list the available servers. Click the system that your report job (<AS/400_Server_Name> – Logic) was sent to and then click the **Select** button. You should see the Submitted Job Search display (Figure 199).

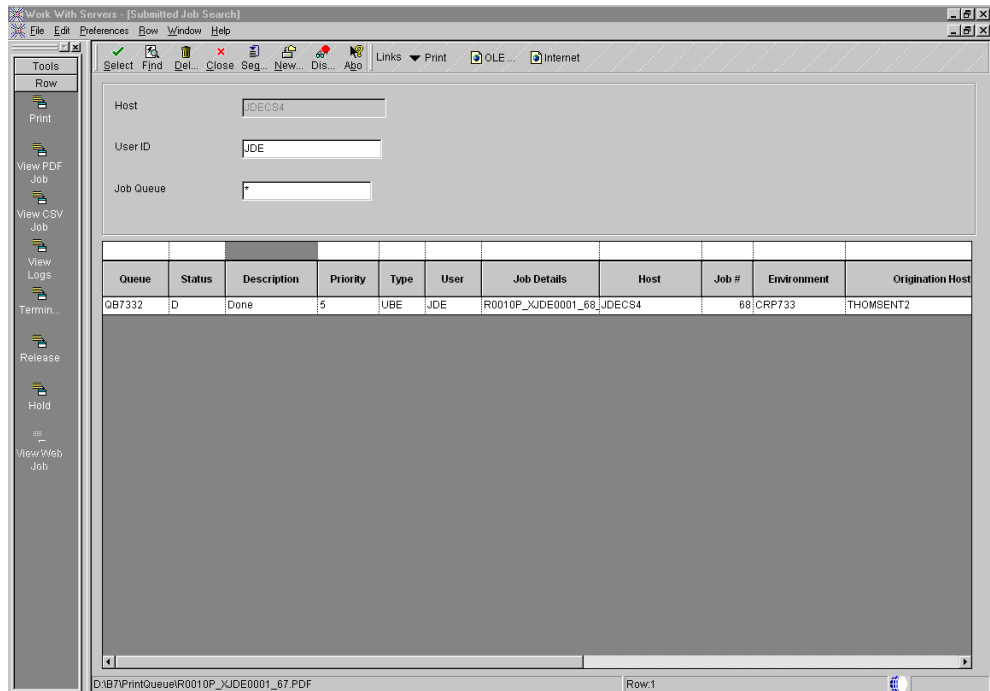
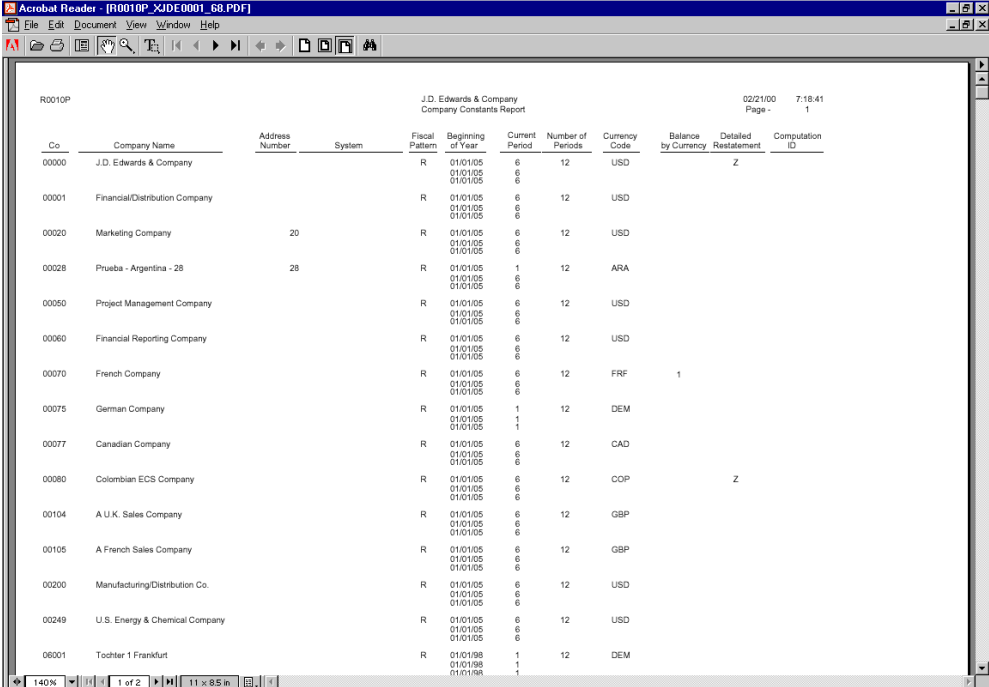


Figure 199. Submitted Job Search display

Note

If your job status is “E” (Error), highlight the job entry. Click **Row**, and select **View Logs**. On the display that follows, select one or both check boxes for Logging (JDE.log) and Tracing (JDEDEBUG.log) to view the associated job logs.

12. From this display, highlight your job, and click the **View PDF Job** icon in the left (Row) toolbar. If you are unsure about which job you submitted, scroll to the right to check the time and date stamp. You should see an Adobe Acrobat Reader session open up, and you can now view your report online as shown in Figure 200.



Co	Company Name	Address Number	System	Fiscal Pattern	Beginning of Year	Current Period	Number of Periods	Currency Code	Balance by Currency	Detailed Restatement	Computation ID
00000	J.D. Edwards & Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD		Z	
00001	Financial Distribution Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
00020	Marketing Company	20		R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
00028	Prueba - Argentina - 28	28		R	01/01/05 01/01/05 01/01/05	1 6 6	12	ARA			
00050	Project Management Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
00060	Financial Reporting Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
00070	French Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	FRF	1		
00075	German Company			R	01/01/05 01/01/05 01/01/05	1 1 1	12	DEM			
00077	Canadian Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	CAD			
00080	Colombian ECS Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	CDP		Z	
00104	A U.K. Sales Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	GBP			
00105	A French Sales Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	GBP			
00200	Manufacturing/Distribution Co.			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
00249	U.S. Energy & Chemical Company			R	01/01/05 01/01/05 01/01/05	6 6 6	12	USD			
06001	Tochter 1 Frankfurt			R	01/01/98 01/01/98 01/01/98	1 1 1	12	DEM			

Figure 200. Report viewed with Adobe Acrobat Reader

13. To print the report to your local printer from this display, select **File->Print** from the toolbar at the top of the display.

14. To print the job through the outq on the AS400 system, highlight the job in Work with Submitted Jobs and select **Print** from the form menu.

For additional assistance, refer to Chapter 5, “Printing OneWorld Reports” in the J.D. Edwards *OneWorld System Administration* manual.

14.3 OneWorld printing rules, tips, and techniques

When setting up J.D. Edwards OneWorld printing, you should know these basic printing rules:

- A printer name must have *no* spaces.
- A printer name can have a maximum of 30 characters.

- A printer name is *case sensitive*.
- The Enterprise Server name is *case sensitive*.
- The J.D. Edwards delivered paper types are Letter, Legal, and A4.
- *Only* the HP9000 uses “ube” in the conversion program string. The AS/400 system and all other platforms use “jdekrnl”.
- Paper trays are not configurable with the JDE printer setup. PostScript and PCL printers automatically know to ask for legal paper or use a legal paper size because part of the PCL or PS file instructs which paper size to use for printing.
- The *host* type in the printer capabilities is just a free-format field, where a descriptive name can be placed of the type of host to which the printer is connected (a required field in B733).
- When defining a default printer, *groups* cannot be used.
- Check printing can be done with a line printer using courier 10 or 11 point fonts. Set grid size to 10 to design the checks.
- Users can pick a printer on the fly versus using a default printer by selecting the Change Printer button on the toolbar when submitting a report.
- G/L posting reports may need to be changed from EVRFIN4 to another printer name.
- You may create a portrait and landscape definition for each printer.
- The report picks up the orientation from the version used.
- If your report batch process prints the wrong font on the report or prints portrait instead of landscape (or vice versa), check the report properties of the version you just ran and verify that they are correct. For assistance, see “Modifying Versions” in the J.D. Edwards *Enterprise Report Writing Guide*.
- If a physical printer is configured, the output queue associated with it on the AS/400 system *must* have the same name.

14.4 Report printing

In J.D. Edwards OneWorld, batch versions for reports are a predefined set of specifications contained in a file that is separate from the base report or batch specifications. These specifications control the logical functions and the appearance of the report. The version contains the processing instructions, which are a complete set of preselected processing options and additional characteristics specific to report design. You can create your own reports or simply modify one of the existing reports or report versions for your business needs.

To modify a predefined report, perform this process:

1. Fast Path to the Batch Versions menu (P98305).
2. Specify a version in the Batch Application field, or click the **flashlight** icon to prompt it. In this example, use R014021 (the One Line Per Address report).
3. Click **Find** to list the versions currently available on your workstation.
4. In the grid, highlight the version that you want to work with, and click the **Advanced** icon from the “Row” menu on the left side of your display. If you do

not see the toolbar on the left side of your display, click **Preferences** on the top of your display, and turn on **Exit Bar**. Your display should appear as shown in Figure 201.

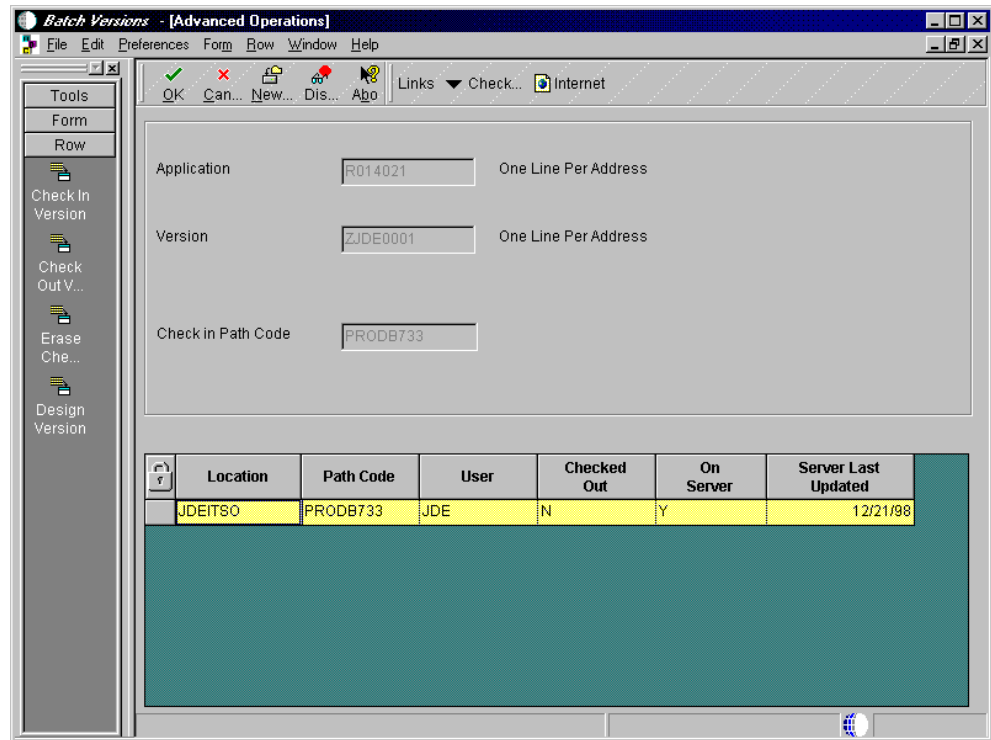


Figure 201. Batch Versions: Advanced Operations display

5. Before you can modify an existing report with Report Design, you must first check out the report version. The check-out procedure copies the specification records from the central objects location (based on your path code) to your workstation. Only one person can check out a batch version at a time.
6. From the Advanced Operations display, select **Check Out Version** from the Row menu. You receive the message:
This operation will overwrite local specifications. Do you wish to proceed?
Select **Yes**.
7. From the Row menu at the Advanced Operations display, select **Design Version**. Report Design opens with the report specifications for the version as can be seen in Figure 202 on page 410.

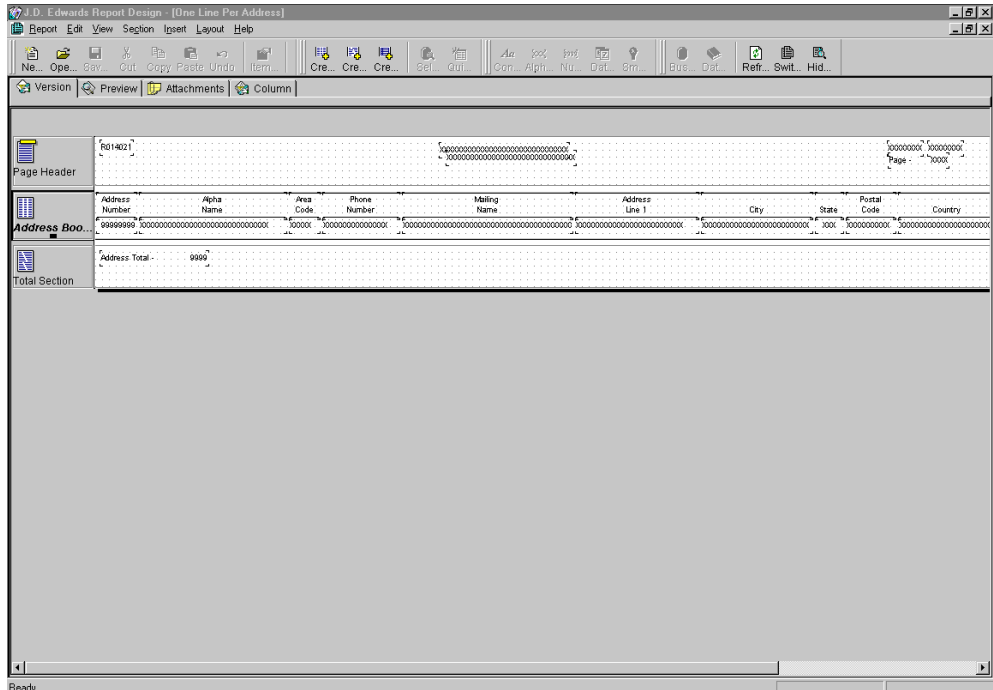


Figure 202. Report Design display

8. Highlight a section (for example, Address Book). Select **Override Version Specifications** from the Section menu in the top toolbar.
 You cannot make any changes to a section until you select the overrides you want to change.
9. Choose any of the following options; **Section Layout, Section Data Selection, Section Event Rules, Section Database Output, Section Sort Sequence**, and select **OK**.
 You can now make changes by double-clicking on columns or selecting options from the Section menu. If you want to suppress a field, for example, double-click the column heading, and select the **Advanced** tab from the properties box. Now click the box next to Visible to remove the check mark, and click **OK**. That field is now omitted from the report.
10. To test your modifications, select the **Preview** tab to generate a sample report. An example is shown in Figure 203.

Address Number	Alpha Name	Area Code	Phone Number	Mailing Name	Address Line 1	City	State	Postal Code	Country
26002	A&E Job			Division 1	345 Main Street	Birmingham	AL	65034	USA
5577	Abbot, Clarice			Clarice Abbot	1407 West Chanango	Englewood	CO	80110	USA
5619	Abbot, Jamie			Jamie Abbot	1407 West Chanango	Englewood	CO	80110	USA
5532	Abbot, Mark			Mark Abbot	1407 West Chanango	Englewood	CO	80110	USA
6002	Abbot, Dominique			Dominique Abbott	1407 West Chanango	Englewood	CO	80110	USA
6044	Abrams, Brooke			Brooke Abrams	10659 South King Street	Denver	CO	80231	USA
7392	Accounts Receivable Department			Accounts Receivable Department					USA
17000	Acme Department Stores			Acme Department Stores	One Ward Road	Denver	CO	80233	USA
17002	Acme Department Store #68			Acme Department Store #68	1730 Holland Drive	Denver	CO	80221	USA
17001	Acme Property Managers			Acme Property Managers	1600 Karen Place	Denver	CO	80333	USA
4272	Aeropuerto Regional			Aeropuerto Regional	Calle 8 No. 33-44	Bogota			Colombia
8571	AR-Regional Movers			Regional Movers	P.O. Box 7184	Denver	CO	80231	United States
6001	Allen, Ray			Ray Allen	410 17th Avenue	Denver	CO	80237	USA
5345	Allen, Cheryl			Cheryl Allen	410 17th Avenue	Fairfax	VA	20151	USA
4360	Allen Supplies			Allen Supplies	9100 E. 105 Street	Denver	CO	80212	USA
281	Almacen Compañia 28 - ESTE			Almacen Compañia 28 - ESTE					USA
283	Almacen Compañia 28 - OESTE			Almacen Compañia 28 - OESTE					USA
4002	Aluminium de Rhone			Aluminium de Rhone	563 Rue de Lisbonne	Lyon		69215	France
8014	Anderson, Jeanette			Jeanette Anderson	8965 W. 17th St.	Denver	CO	86201	United States
8204	Anderson, Jim								USA
5000	A Project Management Company	303	573-3036	A Project Management Company	Stanford Office Center, #200	Parker	CO	80134	USA
5557	Arapahoe County, Clerk of Court			Arapahoe County	Clerk of Court	Aurora	CO	80012	USA
3002	Atlantic Corporation	033	99821806	Atlantic Corporation	96 Rue Anatole France	Paris		92300	France
5844	Ato, Chye Seng			Chye Seng Ato	265 Warren St.	Denver	CO	80231	United States
6832	Ato, Connie			Connie Ato	265 Warren St.	Denver	CO	80231	United States
7372	Auditor for Expense Reimbursement			Auditor for Expense Reimbursement					USA
17003	Bank One Plaza			Bank One Plaza	Bank One Plaza	Denver	CO	80120	USA
				Cindy Ballas	8000 Young St	Thorndell	ON	M9Y 9R8	Canada

Figure 203. Sample report version

Remember, the changes you make only affect the version on your local workstation. To make these changes available to the enterprise, you must check in the version. This has one caveat: Make sure you want to make your changes permanent before you check in a version. To check-in a version, go back to the Advanced Operations screen, and select the **Check In Version** icon from the Row menu.

If you do not want to permanently keep your changes, make sure you erase your check out so that other users can check out this version. At the Advanced Operations menu, select the **Erase Check Out** icon from the Row menu.

For additional assistance, refer to Chapter 12, “Working with Batch Versions” in the J.D. Edwards *OneWorld Foundation* for the B73.3.2 manual. For information about creating custom reports, refer to the J.D. Edwards *Enterprise Report Writing Guide* for B73.3.2.

14.4.1 Printing in batch

In most cases, you may want to submit your reports to batch on the AS/400 Enterprise Server where it can more efficiently handle the processing needs.

When a batch job is sent to run on the AS/400 system, it is referred to as a UBE. To monitor the report batch requests, use the command:

```
WRKACTJOB SBS (QBATCH)
```

Look for jobs with PGM-PRINTUBE in the Function column.

You can run a OneWorld report from your workstation or from an AS/400 command line. To initiate this process, ensure your library list is set up properly and issue the following command:

```
RUNUBE USER (USERNAME) PASSWORD (USERPASSWORD) ENVIRON (CRP733) REPORT (R0006P)
VERSION (XJDE0001)
```

You can also schedule a report to be run on a specific date using the SBMJOB OS/400 command. Here is an example:

```
SBMJOB CMD (RUNUBE USER (USERNAME) PASSWORD (USERPASSWORD) ENVIRON (CRP733)
REPORT (R0006P) VERSION (XJDE0001)) SCDDATE (*FRI) SCDTIME (2330)
```

14.4.2 Print immediate functionality

The workstation jde.ini file settings control whether a report prints immediately and whether OneWorld saves the output after processing the report.

You can open and edit the jde.ini file using Microsoft Wordpad or by using the MS-DOS Editor. Select the **Start** menu in the lower left side. Then choose **Run...** edit c:\WINNT\jde.ini from a Windows workstation.

The PrintImmediate=TRUE/FALSE setting determines whether the system automatically prints the report after processing is complete on the server. The default setting is "False", which means users must use the Work with Servers application to manually print the report. If you specify "True", the system automatically prints the report to the defined printer after it is generated.

The SaveOutput=TRUE/FALSE setting determines whether the system saves or deletes the output after you view or print the job. The default setting is "True", which means the system saves the output after you view or print it. If you select "False", the system deletes it after you view or print it. The print settings are shown in Figure 204.

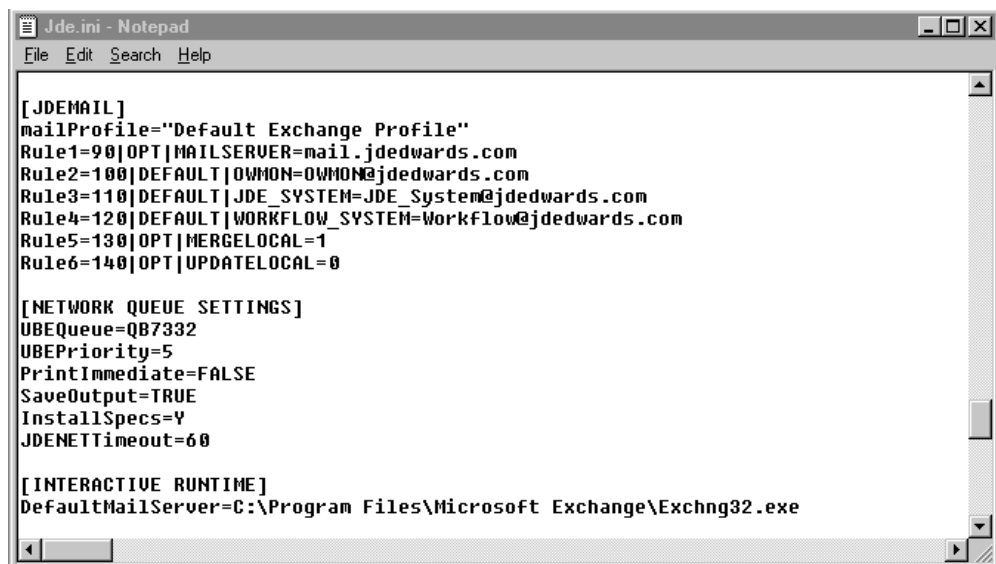


Figure 204. Print settings in jde.ini

14.4.3 Print filters paired to emulation for print stream

Occasionally, you may encounter problems with formatting if the printer driver (*Host print transform*) selected for your printer does not match your printer model. When a remote OUTQ or printer device is configured to use Host print transform, you may define a Manufacturer Type and Model for your printer. You may have to experiment with different model types and try printing from OneWorld to test the output.

Chapter 15. Security

This chapter describes AS/400 security concepts and offers some recommendations on how to setup security in OneWorld environment.

15.1 AS/400 security architecture

System security is an integrated function of the AS/400 system and controls all AS/400 software functions. Users are identified and authenticated by a single security mechanism, at the system level, for all functions and environments available on an AS/400 system, including program development and execution, database applications, office applications, and so forth. All objects on an AS/400 system are under security control, including libraries and files, display stations, operator console functions, programs, menus, and so on. For further information, refer to *OS/400 Security - Reference*, SC41-5302.

15.1.1 Security levels

The AS/400 system has five levels of security controlled by the entry in the QSECURITY system value. An IPL is required to make a change. The security levels are:

- **Level 10:** Level 10 is no longer supported at V4R3.
- **Level 20 (Password):** The system requires a user ID and password for sign-on. All users are given access to all objects.
- **Level 30 (Password and Resource):** The system requires a user ID and password for sign-on. The security of resources is enforced.
- **Level 40 (Password, Resource, and Operating System Integrity):** The system requires a user ID and password for sign-on. The security of resources is enforced. Additional integrity protection features are also enforced.
- **Level 50 (Password, Resource, and enhanced Operating System Integrity):** The system requires a user ID and password for sign-on. The security of resources is enforced. Level 40 integrity protection and enhanced integrity are enforced. Security level 50 is intended for AS/400 systems with high security requirements. It is designed to meet C2 security requirements.

Note

J.D. Edwards OneWorld *is not* currently compatible with Level 50. The recommended level in a OneWorld environment is 40.

15.1.2 AS/400 security system values

The AS/400 system has over 100 variables that control system-wide functions. These are called *system values*. Some of the system values are security-related. These security-related system values fall into four main categories:

- General security defaults
- Audit control
- Password rules
- Other system values related to security

15.1.3 AS/400 user profiles and group profiles

The following terms and concepts are involved in defining users and their authorities to the AS/400 system. Users are defined with profiles and can:

- Be organized into groups
- Have special capabilities
- Have special limitations

15.1.3.1 User profiles

User profiles contain information that describes a system user and that user's privileges and limitations when using the system. The profiles also list the objects that the user owns or is authorized to use. For objects owned by a user, the profile also contains lists of other users' authorizations to those objects.

15.1.3.2 Special authorities

All security systems have special user privileges for certain security and system administration functions. Special authorities allow certain users to administer AS/400 security and system tasks. There are eight special authorities, which are not hierarchical:

- ***ALLOBJ**: All object authority is granted for accessing any system resource.
- ***AUDIT**: Allows the user to perform auditing functions.
- ***JOBCTL**: Allows the manipulation of a job and output.
- ***SAVSYS**: Used for saving and restoring the system and data without having explicit authority to objects queues and subsystems.
- ***SECADM**: Allows the administration of User Profiles and Office.
- ***SERVICE**: Allows access to special service functions for problem diagnosis.
- ***SPLCT**: Allows the control of spool functions.
- ***IOSYSCFG**: Allows the change of system configuration.

15.1.3.3 User classes

There are five user classes that are hierarchical in authority. The classes represent different roles in the IT environment. These are convenient ways to assign the special authorities listed in the previous section to different types of users. The five user classes with their default special authorities include:

- ***SECOFR**: Security Officer (all levels—*SAVSYS, *ALLOBJ*, *JOBCTL, *SERVICE, *SPLCTL, *SECADM, *AUDIT, *IOSYSCFG)
- ***SECADM**: Security Administrator (at security level 20— *ALLOBJ, *SAVSYS, *SECADM, *JOBCTL; at security level 30 or higher—*SECADM)
- ***PGMR**: Programmer (at security level 20—*ALLOBJ, *SAVSYS, *JOBCTL; at security level 30 or higher—no special authorities)
- ***SYSOPR**: System Operator (at security level 20—*ALLOBJ, *SAVSYS, *JOBCTL; at security level 30 or higher—*SAVSYS, *JOBCTL)
- ***USER**: End User (at security level 20—*ALLOBJ, *SAVSYS; at security level 30 or higher—no special authorities)

A user class also affects what options are shown on the system menus. A user with higher authorities sees more of the system menu options. A user with fewer authorities only sees the menu choices allowed by the user class. A user may be

given any of the previously mentioned special authorities regardless of user class. Letting the special authorities be assigned automatically to match the user class is a convenient way to get started.

Special authorities can be assigned specifically, by the security officer or security administrator, when one of the standard user classes does not have the desired combination of authorities.

15.1.3.4 Group profiles

A user profile may be linked to a group profile. This allows all the members of the group to share common attributes, common access to selected objects, and common ownership of objects. A user is not required to be a member of a group. A user may be a member of up to 16 different groups. Group profiles are used to organize users along job functions. The profiles also simplify the assignment and administration of object authorities by authorizing users through a smaller number of group entries. When designing groups, it is important that the group ownership concepts are well understood. It is also important that good naming conventions are used.

A group profile is implemented as a user profile. It is created just like a user profile. In fact, a user profile becomes a group profile when the first user profile specifies it as its group profile. When granting authority, the AS/400 system does not treat groups any differently than user profiles. For easy management, it is better that user and group profiles be used as separate entities. One way to enforce this is to set the group profile password to *NONE. This prevents signon using a group profile.

15.1.3.5 Limited capability

A user may be assigned limited capability. This is done when creating or changing a user profile. Limited capability, when used with an appropriate initial program or initial menu, can restrict a user to a desired subset of the system's functions. Local programming (or using a packaged application) is necessary to accomplish this.

Limited capability (LMTCPB keyword of CRTUSRPRF or CHGUSRPRF commands) may be set to *no*, *partial*, or *full*. The selected value affects the initial program, initial menu, current library, current attention program (associated with the Attention key on the terminal), and access to general system commands.

15.1.4 AS/400 object protection

Since all AS/400 data structures (system and user) are objects, the security system is primarily concerned with protecting objects. All objects have some common structures in their control blocks (invisible to the normal user). This allows a unified approach to security, since all objects interface the same way to the security routines.

15.1.4.1 Authorities

In AS/400 terminology, an authority is the permission to access an object. The object owner and the security officer (or other *ALLOBJ users) can grant or revoke authority to an object. It is important to understand the difference between authority to an object and authority to the data in the object. Operations such as moving, renaming, saving, or deleting apply to the object. It is possible to have authority for these operations without having access to the data stored in the

object. Likewise, you can have full access (read, write, update, delete, execute) to the data in an object without having full authority to manipulate the entire object.

Figure 205 shows the relationship between the object rights and the data rights.

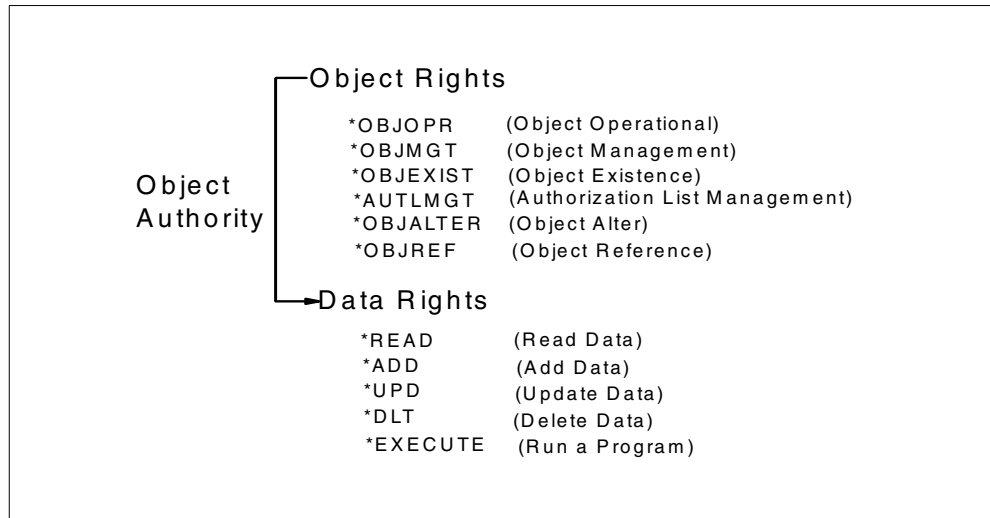


Figure 205. Object authority elements

The following authorities are independent (not hierarchical). For some operations, a combination of authorities is required:

- ***OBJOPR:** The object operational authority controls the use of an object and the ability to look at the description of the object. It is needed to open a file and therefore usually assigned in combination with the desired data rights.
- ***OBJMGT:** The object management authority controls the move, rename, and change attribute functions for the object, and the grant and revoke authority functions for other users or groups.
- ***OBJEXIST:** The object existence authority controls the delete, save, restore, or transfer ownership operations of an object.
- ***AUTLMGT:** This authority is needed to manage the contents of an authorization list associated with the object. This is a specialized security authorization that is not usually grouped with the other seven object authorities.
- ***OBJALTER:** This authority is needed to alter the attributes of database files and change the attributes of SQL packages.
- ***OBJREF:** This authority is needed to specify a database file as the first level in a referential constraint.
- ***READ:** Controls the ability to read data from the object.
- ***ADD:** Controls the ability to insert a new entry (such as a new record in a file) into the object.
- ***UPDATE:** Controls the ability to modify existing entries in the object.

- ***DELETE:** Controls the ability to remove existing entries (for example, records) in the object. Deleting the entire object requires *OBJEXIST authority.
- ***EXECUTE:** Controls the ability to run a program, service program, or SQL package, and to locate an object in a library or a directory.

Some common combinations of authorities were given special names as an abbreviated form. For example, *USE is the combination of *OBJOPR, *READ, and *EXECUTE. Some of the common combinations include:

- ***ALL:** Allows unlimited access to the object and its data.
- ***CHANGE:** Allows unlimited access to the data in the object.
- ***USE:** Allows data in the object to be read.
- ***EXCLUDE:** Allows no access to the object or its data.
- **USER DEF:** User-defined authorities that do not match any of the standard ones.

15.1.4.2 *PUBLIC authority

Public authority is the default authority for an object. It is used if users do not have any specific (private) authority to an object, are not on the authorization list (if one is specified) for the object, or their groups have no specific authority to the object.

15.1.4.3 Authorization lists

An authorization list is an important and commonly used security structure. It is used to authorize a user or a group of users to different types of objects (such as files or programs) secured by the authorization list.

An object may have only one authorization list associated with it. An authorization list may secure more than one object. A user can appear on many different authorization lists. Authorization lists are not affected when objects secured by the authorization list are deleted. If an object is deleted and then restored to the same system, it is automatically linked to an existing authorization list for the object. This is an important advantage of authorization lists.

15.1.4.4 Adopted authority

Certain programs or commands called by a user may require a higher level of authority (for the duration of the command) than is normally available to that user. Adopted authority provides a means for handling this situation. Adopted authority allows a user to temporarily gain the authority of the owner of a program (in addition to the user's own authorities) while that program is running. This provides a method to give a user additional access to objects, without requiring direct authority to objects.

15.1.4.5 Audit Journal

The Security Audit Journal is a facility that allows security-related events to be logged in a controlled way that cannot be bypassed. The security auditing function is managed by the System Values QAUDCTL and QAUDLVL. If you want to turn on auditing for the first time, it may be helpful to use the CHGSECAUD command. Some of the events that may be logged are:

- Authorization failures
- Object creations

- Object deletions
- Changes to jobs
- The moving or renaming of objects
- Changes to system distribution directory or office mail actions
- Obtaining authority from programs that adopt
- System security violations
- Printing actions, both spooled and direct print
- Actions on spooled file data
- Restore operations
- Changes to user profiles, system values, or network attributes
- Use of service tools
- System management functions
- User access to audited objects
- CL command strings

Information from the audit journal can be viewed or printed using the DSPAUDJRNE command. Another method is to go to the Security menu, enter GO SECTOOLS on the command line, and select option 22.

15.2 OneWorld security

OneWorld incorporates AS/400 system security and adds to it in three different ways:

- **OneWorld user profiles and group profiles:** All OneWorld users must have a OneWorld user profile. We recommend that you create a model user profile to use for copying into new user's profiles.

It is not necessary to create an AS/400 user profile by the same name. Multiple OneWorld user profiles can point to a single AS/400 user profile (to simplify database management). This AS/400 user profile, or system user profile, is called the OneWorld security profile.

- **OneWorld security server (KERNEL_DEF4):** The security server kernel runs on a designated OneWorld logic server. When a OneWorld user signs on with OneWorld security server enabled, the security server kernel retrieves the system user profile (from F98OWSEC), which is then passed as a proxy user profile to the AS/400 system. If the OneWorld security server is disabled, the OneWorld user profile and password are passed directly to the AS/400 system. That is to say, the OneWorld user profile and the AS/400 user profile are identical.

Recommendation

Always run with the OneWorld security server enabled. This will enhance your system security. However, it will not prevent third-party software from accessing your database without implementing additional AS/400 database security. This is described in 15.2.5, "OneWorld security recommendations" on page 437.

Apart of enhancing security, the security server allows you to manage access to heterogeneous databases more easily.

- **OneWorld security workbench:** The security workbench application is a powerful tool. It may be used to add, revise, and remove application security.

Note

If you have a coexistent WorldSoftware and OneWorld environment, you need to maintain two independent sets of security files. These products handle security in very different ways. Since we do not address WorldSoftware security in this book, refer to the J.D. Edwards WorldSoftware *Technical Foundation* manual for additional information.

15.2.1 OneWorld user profiles and group profiles

OneWorld uses both user profiles and group profiles to define user environments, user overrides, and application security. If any of the definable values are associated directly with a user profile, these are used for the designated user. The absence of these values for a user profile causes OneWorld to defer to those values defined by a user's group profile.

User profiles define such information as the group to which a user belongs and, optionally, the data source to be accessed. They also define a list of environments that a user or group can select when signing onto OneWorld, the language preference of the user or group, and a certain setup and display features.

After the initial installation, you should begin planning how you want to organize your group profiles and your individual user profiles. A user inherits the rights of the group profile that they are assigned to so the group profiles should be created first. Then, the System Administrator needs to create an individual user profile for every user. Here are the basic steps to complete this task:

1. Before configuring signon security, evaluate whether you want or need all group and user profiles to be setup using the Address Book Revisions program (P01012). This step can be considered optional and is only necessary if you want unique address book numbers for each group and user profile. Certain applications, such as Work Center, require it.
2. Create all of the group profiles for the enterprise. Group profiles are required for each user profile as of B733.2.
3. Create a user profile for every user and assign that user to a group profile.
4. Assign the following preferences to each group or user:
 - *Environments* to determine the environments you want available to each group or user.
 - *Display preferences* to determine OneWorld display characteristics such as language, date format, and country code. The Display Preferences are controlled on the main User Profile Revisions form.

If you are setting up user profiles during the installation process, you must sign onto your Deployment Server using the deployment environment. After you have completed the installation process, you can add or modify user profiles from any machine except the Deployment Server.

To create user profiles, use Fast Path to go to the System Administration Tools (GH9011) menu. Double-click **User Profiles** to select the Work with User Profiles menu (P0092). Your display should appear as shown in Figure 206 on page 422.

From this display, you can add new group or individual profiles, assign environments, and assign preferences. For additional information regarding user profile setup, refer to Chapter 8 in the J.D. Edwards *OneWorld System Administration Guide*.

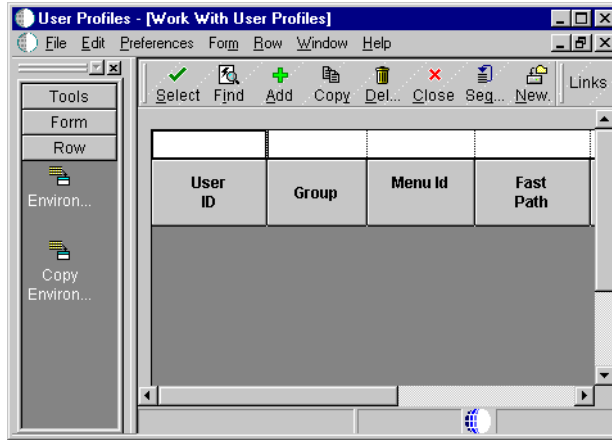


Figure 206. Work with User Profiles display

Note: The OneWorld user profile is created with a *PUBLIC authority of *USE. You can verify this by entering the following command on an AS/400 command line:

```
DSPOBJAUT OBJ(ONEWORLD) OBJTYPE(*USRPRF)
```

The display shown in Figure 207 appears.

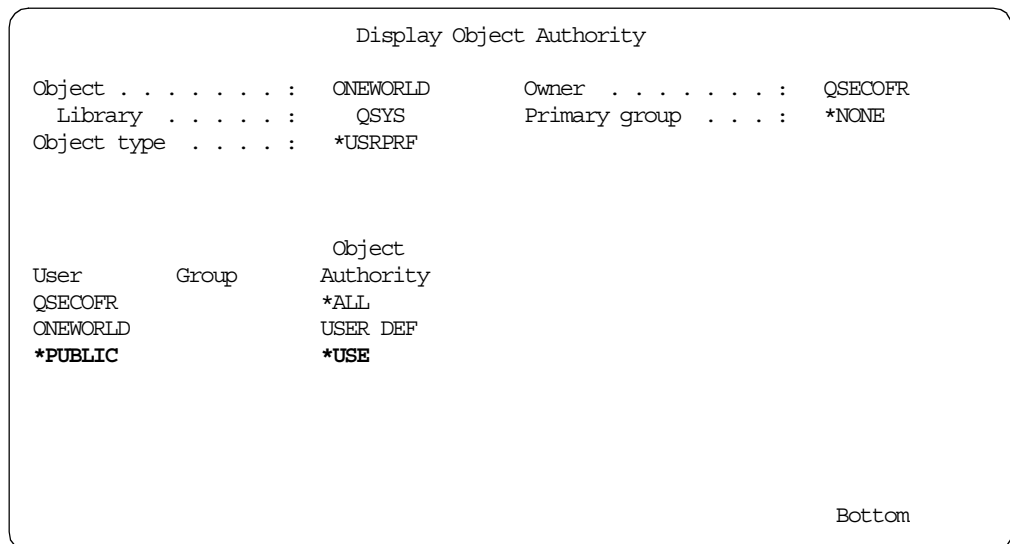


Figure 207. Display Object Authority display

If *PUBLIC does not have *USE authority, you have to issue the command:

```
GRTOBJAUT OBJ(ONEWORLD) OBJTYPE(*USRPRF) USER(*PUBLIC) AUT(*USE)
```

Note

If you choose to run SETOWAUT, the ONEWORLD user profile will exhibit different characteristics. OneWorld administrators will be authorized but *PUBLIC will be changed to *EXCLUDE. See 15.2.5, “OneWorld security recommendations” on page 437.

15.2.2 OneWorld Security Server

The OneWorld Security Server is a kernel used for security validation. It accesses the sign on security table (F98OWSEC) to validate the OneWorld user profile and password, and it retrieves information necessary to access OneWorld databases.

15.2.2.1 Configuring OneWorld user profiles for Security Server

The OneWorld Security Server can be used to define for each OneWorld user profile a corresponding system user profile (security profile). A one-to-one or one-to-many relationship may exist between the system user profile and OneWorld user profile. Additionally, this utility permits the configuration of the data source as a component. Thus, user JDE1 may connect to data source “System - Bnnn” as system user JDEADM and connect to data source “Central Objects - xxxx” as system user JDE.

From the Security Maintenance menu (GH9052), choose **User Security** (P98OWSEC). From that display, click the **Add** button to go to the Security Revisions display (Figure 208).

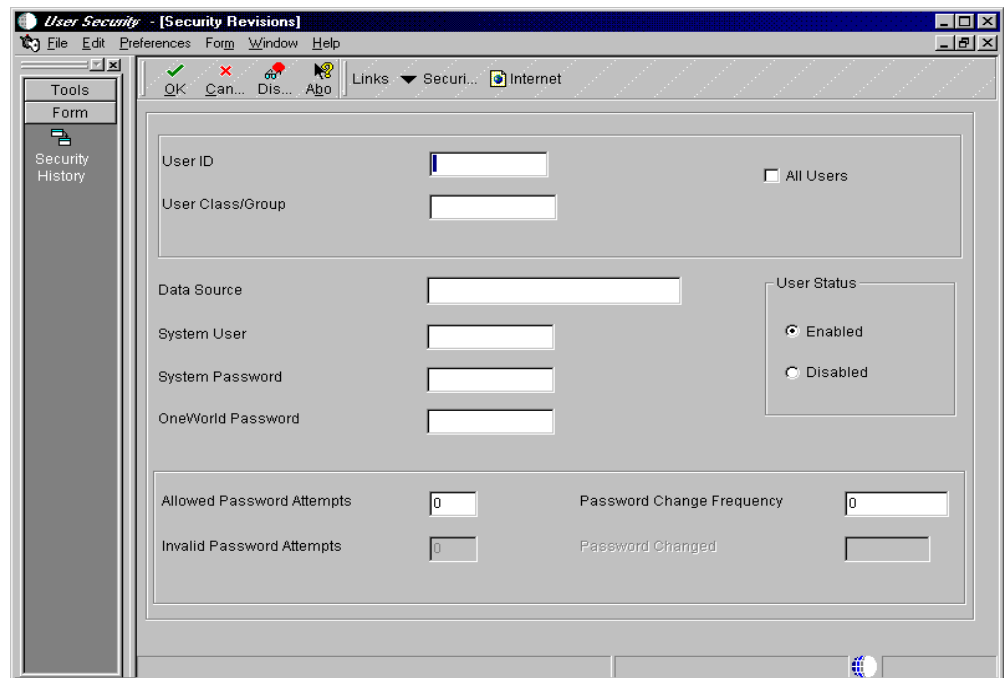


Figure 208. Security Revisions display

If OneWorld Security Server is configured and in use, OneWorld requires user validation at signon and when accessing new data sources. When a user attempts to access an application or perform an action that opens a new

database connection, OneWorld sends a request to the Security Server to gain access to the designated database.

Note

If you need to change a system user password that is used by OneWorld, be sure to end OneWorld services first (ENDNET).

If no security is established at any of these levels, OneWorld lets the user continue. If the OneWorld user is not set up to use Security Server, but attempts to sign on to a workstation configured to use OneWorld Security Server, the user is unable to sign on.

Data source specific

When running OneWorld Security Server, this kernel verifies the system profile and password that has been assigned (to one or more OneWorld users). This system user profile is passed by OneWorld to the database as a proxy in place of the OneWorld user profile. If you leave the data source field blank, it defaults to all data sources, and the word DEFAULT appears in the field. It is possible to assign a different system profile for each OneWorld Data Source by OneWorld user. If the system user ID you are using has different passwords on other servers (based on data source path), create additional entries for your ID with these specific paths (like Central Objects). This can be accomplished by these steps:

1. Fast Path to the Security Maintenance menu (GH9052).
2. Double-click **User Security** (P98OWSEC) to get to the Work with Sign on Security display.
3. Click on the **Find** button to list the user ID. Make a note of which data source (if any) is currently associated with your user ID.
4. Click on the **Add** button to go to the Security Revisions form. Create another entry for your ID with the other Data Source (click the **flashlight** icon to assist with Data Source selection).

Required or not required for Security Server

In the User Security menu (P98OWSEC), there is an option to require the OneWorld Security Server. If you turn this on, a user cannot bypass Sign On security by commenting out their workstation's jde.ini security settings. Once it is enabled, it sets a flag that *every* client workstation *must* be authenticated. It can only be turned off by J.D. Edwards support.

Note

Make sure to test your security set up thoroughly *before* making the Security Server required.

To make Security Server required, follow these steps:

1. Fast Path to the Security Maintenance menu (GH9052).
2. Double-click **User Security** (P98OWSEC) to access the Work With Sign On Security display.

3. Click the **Form** tab in the toolbar on the left side of the display.
4. If you do not see this toolbar, click on **Preferences** in the top toolbar, and select **Exit Bar** from the pull-down menu.
5. Click the **Required/Not Required** icon in the left toolbar. See Figure 209.

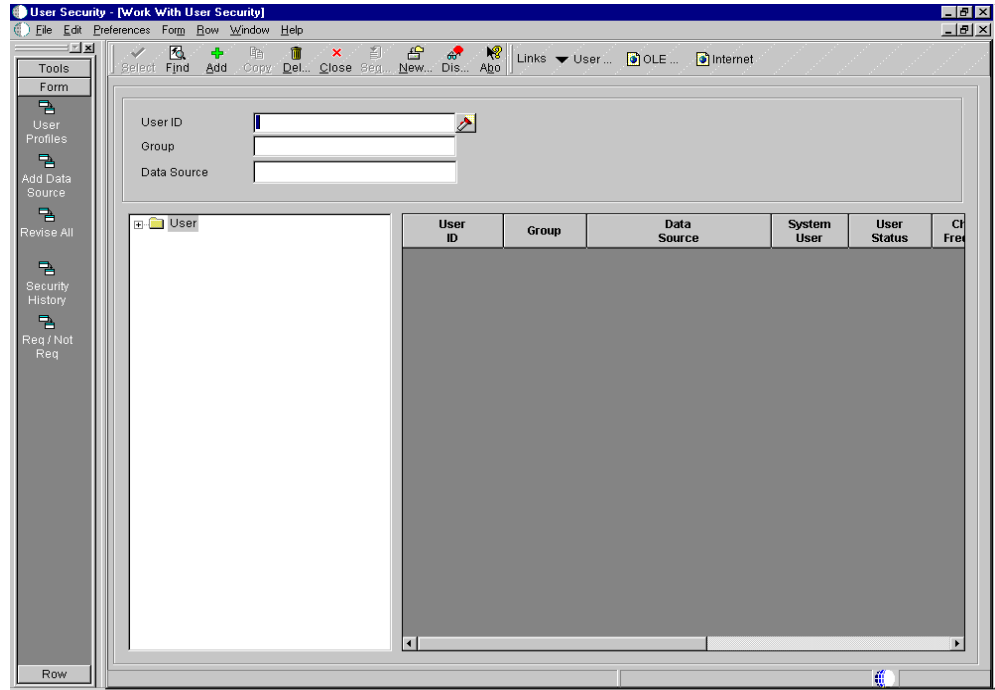


Figure 209. Work with Sign On Security display

6. Click the **padlock** icon to change the status to Security Server - Required. Click the **OK** button to make it take effect. See Figure 210 on page 426.

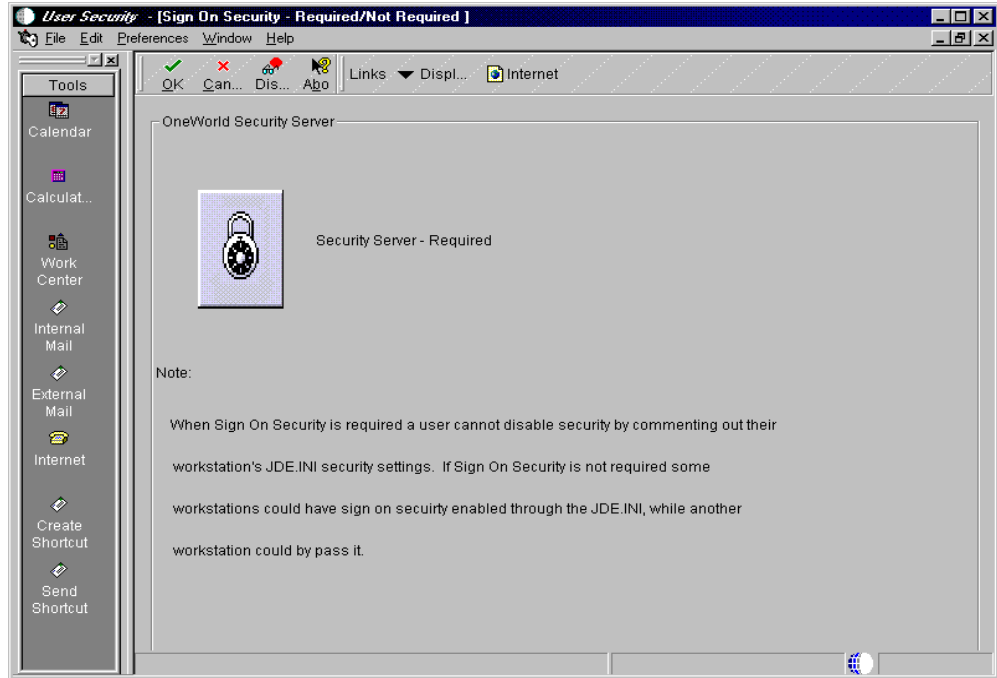


Figure 210. Sign On Security: Required/Not Required form

15.2.2.2 Activating the Security Server

The Security Server validates client signon requests. If you are running multiple servers with replicated data, you can specify up to 10 security servers in case one of the servers goes down. You must ensure that the F98OWSEC table is being replicated, and that it contains the OneWorld and database user IDs and passwords. The client's jde.ini files can specify a primary, secondary, and so on, security server. Thus, if the first server fails, the client workstation can look to the second server for the database password.

Figure 211 shows a sample of a client workstation's jde.ini file with a primary and an alternate security server specified.

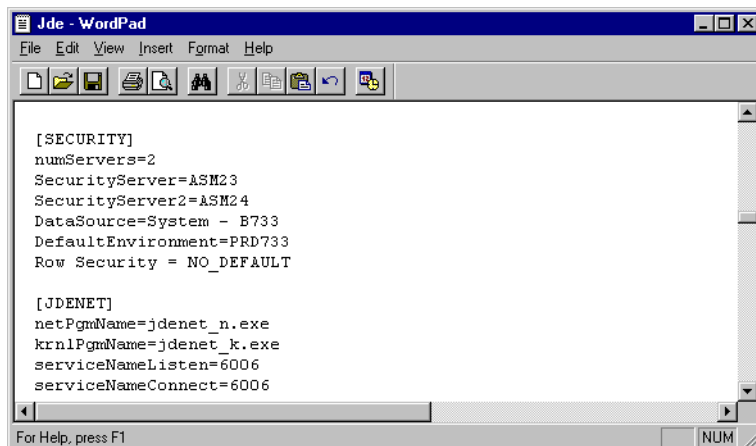


Figure 211. Editing jde.ini file

Setup and configuration

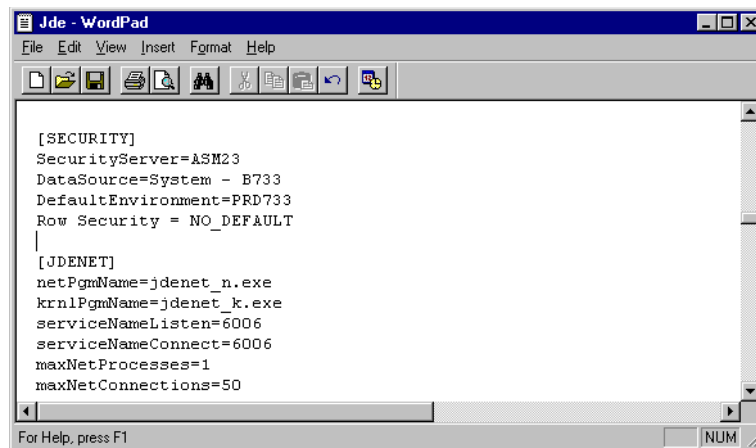
You must modify the Enterprise Server and the workstation jde.ini files to enable and synchronize security settings between the server and workstation. You can enable workstation security by making these changes on the Deployment Server prior to doing package installations on the workstations. If this value is not set properly, the workstations cannot run batch reports on the Enterprise Server.

Perform the following steps:

1. Locate the jde.ini file that will be sent to the workstation as part of a package installation. This file is located on the OneWorld Deployment Server in the release share path \\B733\OneWorld Client Install\Misc\jde.ini.

Note: If you are editing this file on an individual Windows NT workstation, the path is C:\WINNT\jde.ini.

2. Using an ASCII text editor like Microsoft Notepad, view the jde.ini file to ensure the SecurityServer and DefaultEnvironment settings are correct in the SECURITY section. Prior to OneWorld Version B733, the security settings were commented out with a semicolon (;) before each line. See Figure 212.
3. Remove any semicolons from the SECURITY section when you are ready to enable the security server.



```
[SECURITY]
SecurityServer=ASM23
DataSource=System - B733
DefaultEnvironment=PRD733
Row Security = NO_DEFAULT
|
[JDENET]
netPgmName=jdenet_n.exe
krnlPgmName=jdenet_k.exe
serviceNameListen=6006
serviceNameConnect=6006
maxNetProcesses=1
maxNetConnections=50
```

Figure 212. jde.ini file

4. On the AS/400 Enterprise Server, if you have Application Development ToolSet for AS/400 installed (5769-PW1), on a command line, enter:

```
WRKMBRPDM FILE(B733SYS/INI)
```
5. Or run SAW. If you do not have 5769-PW1 installed, you need to FTP the INI file to a workstation, modify it with an ASCII editor, and then FTP it back to the AS/400 system. You should see the Work with Members Using PDM display as shown in Figure 213 on page 428.

```

Work with Members Using PDM
ASM23

File . . . . . INI
Library . . . . . B7332SYS          Position to . . . . .

Type options, press Enter.
2=Edit      3=Copy 4=Delete 5=Display 6=Print 7=Rename
8=Display description 9=Save 13=Change text 14=Compile 15=Create module..

Opt Member      Type      Text
5  JDE

Parameters or command
===>
F3=Exit      F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve   F10=Command entry  F23=More options  F24=More keys

Bottom

```

Figure 213. Work with Members Using PDM display

6. From this display, type 5 on the *Opt* field next to JDE, and press Enter.
7. On the next panel, type SECURITY on the SEU==> line on the top of the screen, and press the F16 key until you see the display that appears in Figure 214.

```

Columns . . . . : 1 71          Browse          B733SYS/INI
SEU==>
FMT **  .... 1  .... 2  .... 3  .... 4  .... 5  .... 6  .... 7
0234.00 [SECURITY]
0235.00 DataSource=System - B733 - DNT
0236.00 User=JDE
0237.00 Password=JDE
0238.00 DefaultEnvironment=PRD733
0239.00 SecurityServer=ASM23
0240.00 ServerPswdFile=TRUE
0241.00 History=0
0242.00
0243.00 [LOCK MANAGER]
0244.00 Server=RCHASM23
0245.00 AvailableService=NONE
0246.00 RequestedService=NONE
0247.00
0248.00 [TCENGINE]
0249.00 TraceLevel=0
0250.00 StopAfterRow=0

F3=Exit  F5=Refresh  F9=Retrieve  F10=Cursor  F11=Toggle  F12=Cancel
F16=Repeat find      F24=More keys

```

Figure 214. Enterprise Server jde.ini file

8. Verify that the Security Server settings are correct. You should also make sure that the INI file is secured.

15.2.2.3 Encrypting security server passwords

When you turn on the security server, you will want to encode user passwords for batch jobs. You need to change the settings in the [SECURITY] section in the B733SYS/INI file. See Figure 214. To encode passwords, perform these tasks:

1. Enter the following statement:

```
STRSEU B733SYS/INI JDE
```

2. Modify the following setting:

```
[SECURITY]  
ServerPswdFile=True
```

Note

To deactivate encoding, you should change the settings to `FALSE`. This is *not* recommended if you are running with the security server turned on.

15.2.3 OneWorld Security Workbench

The Security Workbench application is a powerful tool for security administrators to control OneWorld security. The security attributes are configured through various forms, but they are stored as records in just one table, F00950. From this single application, you can set up virtually all OneWorld security types. You need to limit access to the Security Workbench so only one or two individuals have access to it. All changes should be routed through these individuals.

Here is a list of the different OneWorld security types:

- **Application Security:** Secures users from running or installing a particular application or a particular form within an application.
- **Action Security:** Secures users from executing a particular action, such as adding, deleting, revising, inquiring, or copying a record.
- **Row Security:** Secures OneWorld users from accessing a particular range or list of data in any table. We recommend that you use row security sparingly because it can have an adverse affect on performance. Cost center security is implemented using row security.
- **Column Security:** Prevents users from viewing a particular field or changing the value for a particular field. It can be a database field or a field that is defined in the Data Dictionary but is not in the database.
- **Processing Option Security:** Secures users from changing the values of processing options or from prompting for versions and prompting for values for specific applications.
- **Tab Security:** Prevents users from seeing or using a tab (or tabs) on a given form.
- **Exit Security:** Secures users from menu bar exits on OneWorld forms and also provides restrictions to the hyper button.
- **Exclusive Application Security:** Sets security specific to an application regardless of any other security that may be set. When you set exclusive application security for a user, that particular user gains access to just the specific application that you define. All other security still applies.

- **External Calls Security:** Secures users from accessing stand-alone executables that exist external to OneWorld (for example, design tools, system monitors, and debugging tools).
- **ActivEra:** Sets security for eight ActivEra features:
 - ActivEra Portal
 - Task Documentation
 - Fine Cut
 - Favorites
 - Effectivity dating (date of release)
 - ActivEra Explorer
 - Rough Cut
 - Universal Director

To access the Security Workbench, Fast Path to the Security Maintenance menu (GH9052). Your display should appear as shown in Figure 215.

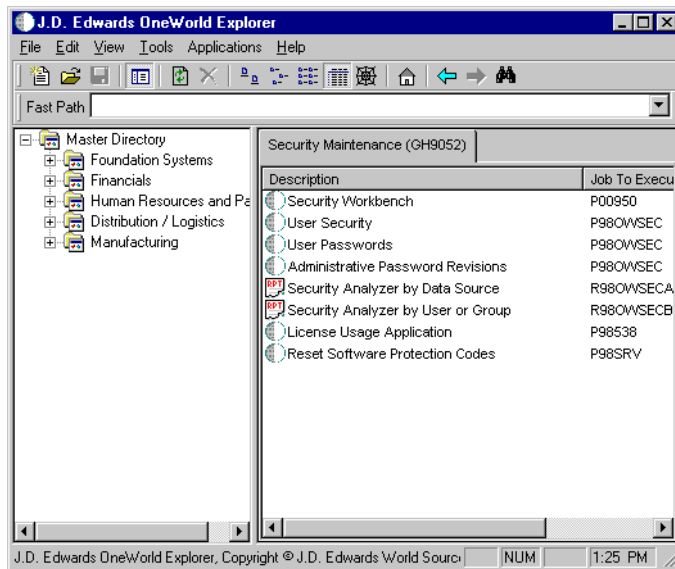


Figure 215. Security Maintenance menu

From this display, double-click **Security Workbench** (P00950) to select it. Your new display should appear as shown in Figure 216.

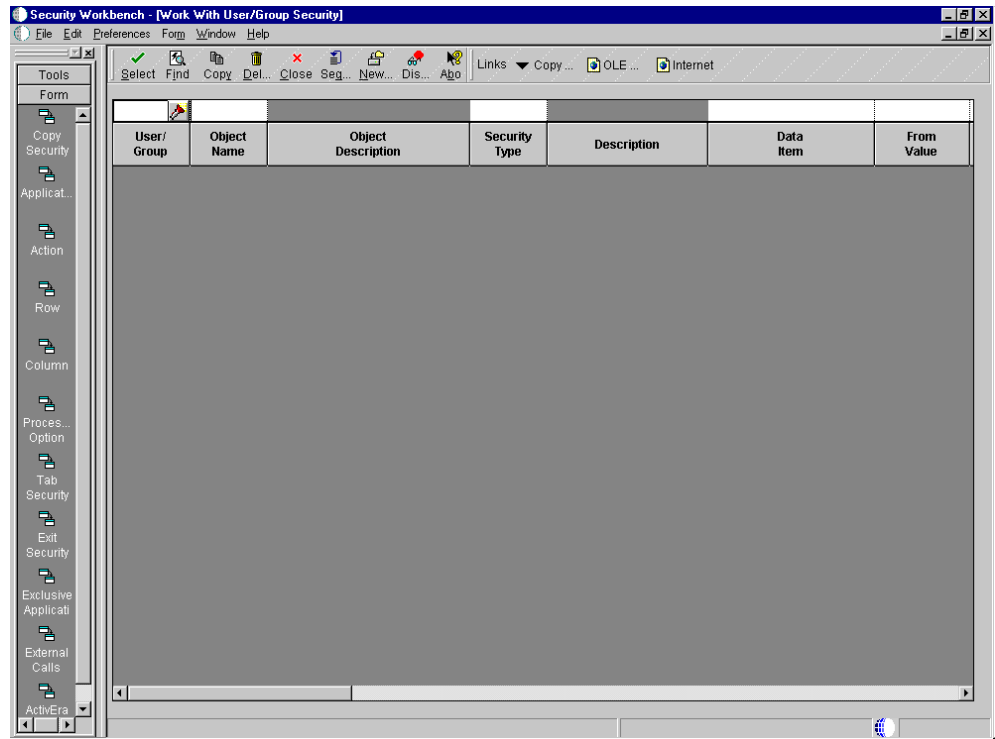


Figure 216. Security Workbench menu

From this menu, you can access all of the forms to tailor the different OneWorld security types by selecting one of the icons in the Form toolbar. If you do not see the toolbar on the left-hand side of your display, click **Preferences** on the top toolbar. Then, select **Exit Bar** from the pull-down menu.

15.2.3.1 Setting up Row Security

To set up row security, complete these steps:

1. Fast Path to the Data Dictionary Design menu (GH951).
2. Choose **Work With Data Dictionary Items** (P92001).
3. On the next display, click the **Find** button to view the data items. Highlight the data item that you want to secure, and click the **Select** button.
4. On the Data Item Specifications display, select the **Row Security** check box. Your display should appear as shown in Figure 217 on page 432.

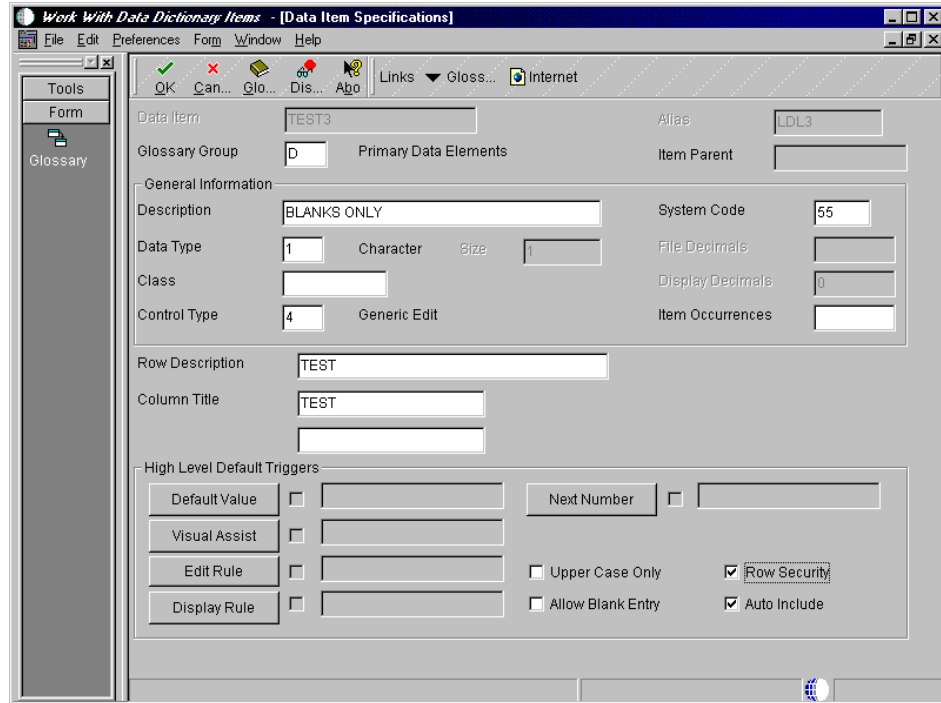


Figure 217. Data Item Specifications display

5. Click the **OK** button to accept this change.
6. Fast Path to the Security Maintenance Menu (GH9052), and choose **Security Workbench** (P00950).
7. On the Work With User/Group Security form, select the **Row** icon from the left Form toolbar. Now you can tailor table row security from the Row Security Revisions form.

Remember, OneWorld caches certain security information from the Security Workbench table (F00950) in the workstation's memory cache. If system administrators make changes to that table, users must log off and then log back on before the changes affect them.

15.2.3.2 Menu security and fast path

Menu security limits user access to specific menus. In WorldSoftware, you can set up menu masking to secure entire menus or individual selections on a menu by user. In OneWorld, menu security is accomplished by specifying an initial, menu for the group or user in the User Profile Revisions program. Then the administrator can restrict users from having Fast Path access to other menus. Fast Path is typically allowed for application team leads, not end users.

In OneWorld, you can think of menus as existing in a tree kind of structure. When you give a user access to a menu, OneWorld also allows access to all menus that branch out lower from the original menu. The user is not allowed, via menus, to go higher up in the tree structure. To complete menu security, you need to restrict the user's Fast Path access to the higher menus as well.

To obtain menu security and Fast Path access, perform these tasks:

1. From the System Administration Tools menu (GH9011), select **User Profiles** (P0092).
2. From the Work with User Profiles display, click the **Find** button to view the User IDs.
3. Highlight the user ID that you want to modify, and click the **Select** button.
4. From the User Profile Revisions display (Figure 218), you can define the initial menu (Menu Identification) for each user and whether they are allowed to Fast Path.

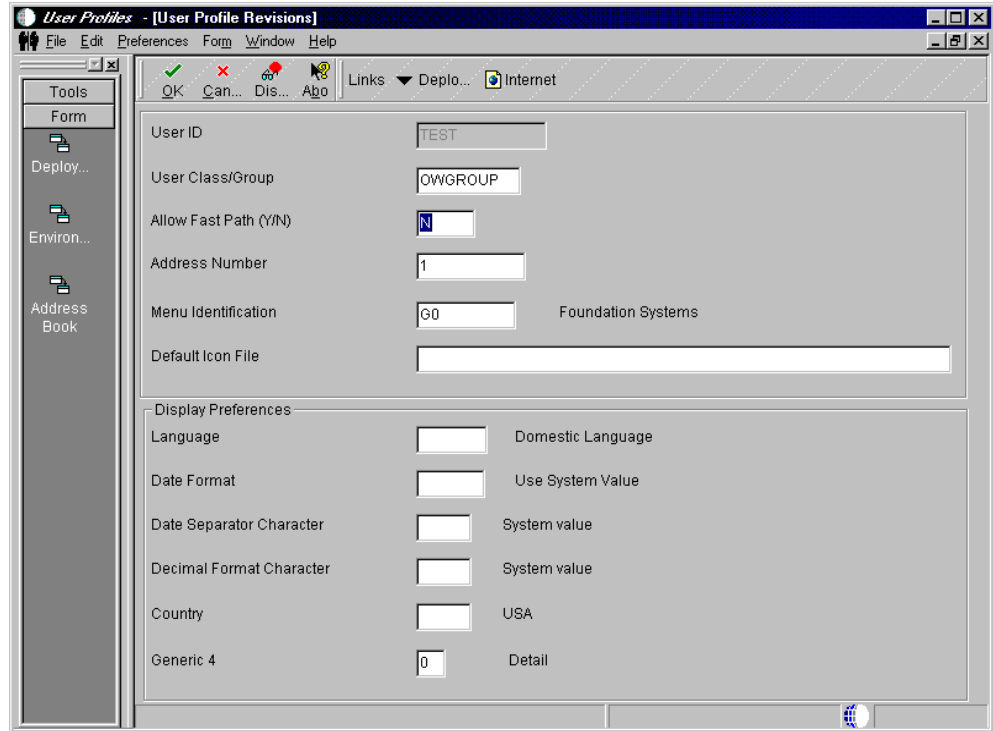


Figure 218. User Profile Revisions display

Most application team leads should have access to UDCs, but users should not. For Data Dictionary, the DD administrator should be the only one making changes to maintain tight control of it.

Customers who are using Workflow when automating their business processes should also implement Workflow Security. A Workflow administrator or Application lead should monitor and secure Workflow queues and applications. For additional Security Workbench assistance, refer to Chapter 9 in the J.D. Edwards *OneWorld System Administration Guide*.

15.2.4 Deployment Server directory and file security

You may want to set up Windows NT security on the OneWorld directories and files on your Deployment Server.

15.2.4.1 Directories

After your OneWorld installation is complete, and you have successfully installed at least some client workstations, you can optionally set up Windows NT security

on the OneWorld directories and files on your Deployment Server. To set up Windows NT security, complete the following steps:

1. Sign on Windows NT on the Deployment Server as *Administrator*.
2. Using Windows NT Explorer, locate and right-click the directory that you want to secure.
3. Click **Properties**.
4. Click the **Security** tab.
5. Click **Permissions**.
6. Click **Replace Permissions on Subdirectories**.
7. Click **Replace Permissions on Existing Files**.
8. Select the particular user or group and select the required authority, based on Table 25 through Table 34 on page 437 and click **OK**.

Use Table 25 through Table 34 on page 437 to determine what permissions are required for the various types of OneWorld users. These tables contain updates to the similar tables listed in the *OneWorld Installation Guide (AS/400 Systems)*.

The directory names listed in the table captions of Table 25 through Table 34 on page 437 are subdirectories of the \JDEdwardsOneWorld\B733 directory. You can apply the indicated permissions to all subdirectories and files except where noted.

Table 25 contains recommended user permissions for the OneWorld client directory.

Table 25. *OneWorld client directory authority*

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	Read Only (all files except jdeclnt.ddc and jdeclnt.xdc) Change (jdeclnt.ddc and jdeclnt.xdc)
Development users	Read Only (all files except jdeclnt.ddc and jdeclnt.xdc) Change (jdeclnt.ddc and jdeclnt.xdc)
CNC administrators and application leads	Read Only (all files except jdeclnt.ddc and jdeclnt.xdc) Change (jdeclnt.ddc and jdeclnt.xdc)

Table 26 contains recommended user permissions for the OneWorld path code directory.

Table 26. Path code (CRPB733, DEVB733, PRDB733, PRTB733) directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	Read Only (all directories and files except directories \path code and \package) Change (directories \path code and \package)
Development users	Change
CNC administrators and application leads	Change

Table 27 contains recommended user permissions for the OneWorld database directory.

Table 27. OneWorld database directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	No Access
Development users	No Access
CNC administrators and application leads	No Access

Table 28 contains recommended user permissions for the OneWorld Data Dictionary directory.

Table 28. OneWorld AS/400 Data Dictionary library (DDB733) authority

User type	Permissions
JDE	Change
Production users	No Access
Development users	No Access
CNC administrators and application leads	Change

Table 29 contains recommended user permissions for the OneWorld helps directory.

Table 29. OneWorld helps directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	Read Only
Development users	Read Only
CNC administrators and application leads	Read Only

Table 30 contains recommended user permissions for the OneWorld hosts directory.

Table 30. OneWorld hosts directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	No Access
Development users	No Access
CNC administrators and application leads	No Access

Table 31 contains recommended user permissions for the OneWorld mediaobj directory.

Table 31. OneWorld mediaobj directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	Read Only
Development users	Read Only
CNC administrators and application leads	Change

Table 32 contains recommended user permissions for the OneWorld planner directory.

Table 32. OneWorld planner directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	No Access
Development users	No Access
CNC administrators and application leads	No Access

Table 33 contains recommended user permissions for the OneWorld PrintQueue directory.

Table 33. OneWorld PrintQueue directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	No Access
Development users	No Access
CNC administrators and application leads	No Access

Table 34 contains recommended user permissions for the OneWorld system directory.

Table 34. OneWorld system directory authority

User type	Permissions
Everyone	Read Only
JDE	Change
Production users	Read Only
Development users	Read Only
CNC administrators and application leads	Change

15.2.5 OneWorld security recommendations

J.D. Edwards recently completed a security project called SETOWAUT, which provides an additional set of security tools for OneWorld. We recommend that you secure your system using SETOWAUT tools. The tools and their accompanying white paper “OneWorld: AS/400 Database Security, Release B73.3.2” can be found on J.D. Edwards’ Knowledge Garden Web site at:

<https://knowledge.jdedwards.com>

To access SETOWAUT tools, click **Product->Software Updates**. Then request 3360546 as the Software Update Number.

Since SETOWAUT may have periodic updates, continue to check the Knowledge Garden for refreshes.

15.2.5.1 Why use SETOWAUT

SETOWAUT offers a comprehensive strategy to complete your OneWorld security implementation. After OneWorld is initially installed, the following points are true:

- All OneWorld objects are owned by user profile JDE.
- *PUBLIC has *ALL authority to all OneWorld objects.
- JDE and ONEWORLD are required profiles.

SETOWAUT builds on this basis by securing profiles and objects for OneWorld on the AS/400 system.

In a non-coexistence environment (WorldSoftware is *not* on the same system as OneWorld), SETOWAUT controls authority for all non-system OneWorld libraries at the library level with authorization list ONEWORLD. SETOWAUT does not affect existing authority for individual objects in those libraries. However, the System Library is authorized by the OWADMINL authorization list, and its objects are authorized either by the OWADMINL or ONEWORLD authorization lists.

Note

The default name for System Library for OneWorld B733.2 is B7332SYS.

Table 35 offers a summary of the settings that SETOWAUT implements for you.

Table 35. SETOWAUT security settings

Library	Description of security
Non–Coexistence: all libraries except the System Library (B7332SYS)	The JDE user profile owns datapath libraries and the ONEWORLD user profile owns the object libraries. All objects within these libraries remain unchanged, which means that either JDE or ONEWORLD owns the objects and *PUBLIC has *ALL authority to these objects. The library authority to create objects within the library is set to *ALL (*PUBLIC will have *ALL authority).
Coexistence: all libraries except the System Library (B7332SYS)	All libraries are the same as in a non–coexistence environment except an additional profile, such as JDEGRP, is added to secure datapath libraries. This profile has *USE authority to the library. For WorldSoftware, the SBMJOB command requires *USE authority to all the libraries in the library list to function properly. The JDE user profile, for table objects, owns all of these objects. Each datapath object is secured by *AUTL ONEWORLD.
OneWorld System Library (B7332SYS)	The OneWorld System Library, B7332SYS, is owned by the ONEWORLD user profile and its *PUBLIC authority is set by the *AUTL OWADMINL. All objects associated with OneWorld server administration are secured with the *AUTL OWADMINL authorization list. All other objects are secured by the *AUTL ONEWORLD. Administrative programs, such as CLRIPC, STRNET, ENDNET, PORTTEST, among others, are set to adopt the authority of the owner.

We recommend that you run SETOWAUT immediately after your installation if possible and whenever you have made changes to your OneWorld environments. You may run SETOWAUT at a later time as well. At a minimum, you need to evaluate your security each time after loading OneWorld Service Packs, cumulative updates, ESUs, ASUs, and so on.

Although we recommend that you run SETOWAUT if you have a coexistence environment, this section does not cover SETOWAUT for coexistence.

Note

If you are using WorldSoftware in a coexistence environment, download a document number 3385954 for the WorldSoftware Security setup procedures. This should be implemented before SETOWAUT. Also, review the SETOWAUT documentation (3360546) for coexistence examples.

15.2.5.2 Setting up database security with SETOWAUT

Setting up SETOWAUT is comprised of three steps:

1. Running the CRTIPCS program
2. Running SETOWAUT
3. Verifying your results

The SETOWAUT authority settings are implemented only on the AS/400 system where the command is executed. To avoid having an object locked when you are

trying to change its authority, especially with data sources, we recommend that you run SETOWAUT when OneWorld is down (ENDNET).

Running the CRTIPCS program

The function of the CRTIPCS program is to create a new version of the IPCS program. The function of the IPCS program is to remove IPCS (Interprocess Communication Structures) objects that were created previously by an instance of OneWorld running on the logic server. There are a number of different types of IPCS objects such as semaphores, shared memory, and message queues. OneWorld uses them for communicating and synchronizing between processes to lock, share, and pass information. For more information on Interprocess communication, refer to Appendix D, "Interprocess communication (IPC)" on page 821.

1. Sign onto the AS/400 system as a user with *ALLOBJ and *SECADM authority.

2. Enter the following command:

```
ADDLIBLE SETOWAUT
```

3. Enter the following command for your System Library:

```
ADDLIBLE B7332SYS
```

4. In the SETOWAUT library, on the command line, enter the following command for your System Library:

```
CRTIPCS TGTLIB(B7332SYS) TGTRLS(release)
```

Here, *release* represents the target OS/400 release of the OneWorld objects, for example, V4R2MO.

CRTIPCS renames the existing IPCS program to IPCSOLD and uses the SETOWAUT IPCS module to create a new IPCS program. In a future OneWorld release, CRTIPCS may not be required.

Running the SETOWAUT tool

The following steps describe how to run SETOWAUT:

1. Type the following command:

```
SETOWAUT
```

Press F4, and then press F11. The Set Up OneWorld Authority screen appears (Figure 219 on page 440).

```

Setup OneWorld Authority (SETOWAUT)

Type choices, press Enter.

FULL, DTAPATH, PROF, SYSTEM . . .          *FULL, *PROFILE, *DTAPATH...
INI Library . . . . .                      *NONE, Library
Datapath (library) . . . . .              *INI or LIBRARY1, LIBRARY2...
      + for more values
User Profile Options:
  Modify ONEWORLD Profile . . . Y          Y N
  Modify JDE profile . . . . . Y          Y N
  Modify Security Profile . . .           JDEOW, JDEFIN...
      + for more values
  OneWorld DB Admin Profile . .          *INI, User Profile
  Libs or *INI (Default PathCode)        *INI or NAME1, NAME2...
      + for more values
  Secure Log Path . . . . . N             Y N
  Secure Objects For Coexistence         *NONCOEXIST *NONCOEXIST, *COEXIST...

```

Figure 219. SETOWAUT command

2. Complete the fields and options as necessary, and press Enter. The parameters are explained here:

- **TYPE:** Depending on the value you enter in this field, you can implement a full security setup, modify only the security profiles, or modify only the datapaths' authority. A full security setup includes the System Library, datapath, path code, and user profiles. The list of values are:
 - **FULL:* Use *FULL when you initially implement SETOWAUT. This directs SETOWAUT to perform all the security routines. *FULL is required to be run at least once.
 - **DTAPATH:* Use *DTAPATH only if you need to secure a datapath.
 - **PROF:* Use *PROF to perform only the user profile routines. SETOWAUT will use the user profile settings in the command to direct the process.
 - **SYSTEM:* Use *SYSTEM to perform the System library authority functions. This secures your System Library (B7332SYS), and all of the objects contained within, with the *AUTL OWADMINL. Additionally, all the *PGM objects with attributes of *CLP, *CLLE, or *CLE will have the program attributes modified for adopt authority. The System Library (B7332SYS) is treated differently to allow administration of OneWorld.

You can use this parameter to lock other non-system libraries that contain objects that you use to administer OneWorld.

Note

Use caution when you lock a library with OneWorld security that contains third-party software. J.D. Edwards does not support AS/400 OneWorld database security with third-party software.

- **INILIB (INI Library):** This field identifies to the security application the library in which the jde.ini file resides. The *NONE value allows you to specify that the jde.ini file is either not needed or not available.

- Use a library name, if the following are *all* true:
 - If there is a OneWorld INI library located on the host system
 - If the control files (OCM) are located on the host system
 - If the jde.ini file references the OCM library
- If type = *FULL or *SYSTEM, the library and all the objects are secured with SYSTEM attributes.

SETOWAUT uses the jde.ini file to perform all the INI retrievals. If there is no INI and the type is *SYSTEM, SETOWAUT secures only the library entered.

- Use *NONE, if any of the above requirements is false. This requires actual values in any parameter that allows the value *INI.

- **DTAPATH [Datapath (library)]:** Type *INI in this field to use the datapaths set in the jde.ini file. You can also type specific datapaths in this field. You can type up to 10 data paths at a time. If you use *INI, SETOWAUT modifies each library based upon the ALLOBJECTS parameter.

Note

If you set the INI library field to *NONE, you must manually set datapaths in this field.

Use *INI when the INILIB parameter contains the library name where the jde.ini file is located (INILIB value is not *NONE). This parameter tells SETOWAUT to use the jde.ini file to retrieve the datapath libraries. SETOWAUT retrieves the library name from the jde.ini value in '[DB SYSTEM SETTINGS] Library' and uses this setting to access the OCM files (F986101 and F98611). SETOWAUT selects all the library names (F98611.OMLIB) that meet the following criteria: F986101.OMDATP = F98611.OMDATP, OMUGRP = *PUBLIC, OMSTSO = 'AV', OMSRVR = *the host name*.

- **Modify ONEWORLD profile:** ONEWORLD is a required profile. A value of Y or N is valid for this field.

Use Y to modify or create the ONEWORLD profile that has not yet been modified. ONEWORLD user profile parameters are set as follows:

- GRPPRF set to *NONE
- SUPGRPPRF set to *NONE
- USRCLS set to *USER
- INLMNU set to *SIGNOFF
- INLPGM set to *NONE
- SPCAUT set to *JOBCTL

In addition, SETOWAUT makes the following authority changes:

- Grants authority to profile ONEWORLD to *USE profile QSECOFR.
- Revokes *ALL authority from *PUBLIC (changes it to *EXCLUDE).

Use N only if the profile ONEWORLD already has the correct attributes.

- **Modify JDE profile:** JDE is a required profile. A value of Y or N is valid for this field.

Use Y to modify or create the JDE profile that has not been modified. JDE user profile parameters will be set as follows:

- GRPPRF set to *NONE
- SUPGRPPRF set to *NONE
- USRCLS set to *USER
- INLPGM set to *NONE
- SPCAUT set to *JOBCTL *SAVSYS

In addition, *ALL authority for JDE profile will be revoked from *PUBLIC (changed to *EXCLUDE).

Use N only if the profile JDE has the correct attributes.

- **Modify Security profile:** Security Profile is an AS/400 user profile. If you always run OneWorld Security Server kernel, as recommended, this is the system user profile that the kernel passes to the AS/400 database as a proxy in place of the OneWorld user profile.

Note

We recommend that you delete existing OneWorld Security Profiles prior to running SETOWAUT. After running SETOWAUT and creating OneWorld Security Profiles, the passwords must be changed to correspond with passwords that were set up using OneWorld User Security. The Security user is used as the System user in OneWorld User Security. See Sample Results of SETOWAUT for more information about specific profile properties.

You can enter up to 10 security profiles at a time in this field to modify with the SETOWAUT program. If you enter a security profile that does not already exist, SETOWAUT creates the profile and modifies the profile accordingly.

The security profiles you enter here are created or modified in the following ways:

- USRCLS set to *USER
- INLMNU set to *SIGNOFF
- INLPGM set to *NONE
- SPCAUT set to *NONE
- GRPPRF set to ONEWORLD
- SUPGRPPRF set to JDE

In addition, SETOWAUT makes the following authority changes to the security profile(s):

- Revokes *ALL authority from *PUBLIC.
 - Grants profile ONEWORLD *CHANGE authority to security profile(s).
 - Grants security profile(s) *CHANGE authority to ONEWORLD.
- **OneWorld DB Administrator profile:** If you type *INI in this field, SETOWAUT retrieves the user and password values from the [SECURITY] section in the jde.ini file. If you type a value in this field that does not exist, SETOWAUT creates a profile, the password which is the same as the profile name. If the profile exists, SETOWAUT modifies the profile to be a OneWorld database administrator.

Enter a profile to be used as a database administrator. This profile will have all rights to all OneWorld objects. These database administrator profiles

are allowed to perform certain OneWorld processes (RUNUBE and PORTTEST) that a normal administrator cannot perform.

If the profile does not exist, it creates the profile with the password the same name as the profile. Also, if the profile does not exist, it sets the password to expire (PWDEXP = *YES). DB administrator user profile parameters are set as follows:

- If BV3C is in a library list, SETOWAUT places this program as the initial program (this program will list all the OneWorld occurrences to allow the user to select one occurrence at signon).
- USRCLS set to *PGMR
- SPCAUT set to *NONE
- GRPPRF set to ONEWORLD
- SUPGRPPRF set to JDE

In addition, SETOWAUT makes the following authority changes:

- Revokes *ALL authority from *PUBLIC
- Grants ONEWORLD *USE rights to DB ADMIN profile
- **BSFNLIB:** Type *INI in this field to use the path code library and the associated specification file directory set in the jde.ini file. Use *INI when the INILIB parameter contains the library name where the jde.ini file is located (INILIB does not contain *NONE). This parameter tells SETOWAUT to use the jde.ini file to retrieve the application path code libraries. SETOWAUT retrieves the library name from the jde.ini value in “[DB SYSTEM SETTINGS] Library” and uses this setting to access the OCM files (F986101 and F98611). SETOWAUT selects all the library names (F98611.OMLIB) that meet the following criteria: F986101.OMDATP = F98611.OMDATP, OMUGRP = *PUBLIC, OMSTSO = ‘AV’, OMDBNM = F00942. SETOWAUT will retrieve EMPATHCD (path code) from each record in the F00942 for each library (F98611.OMLIB).

Note

If you set the INI library field to *NONE, you must set path codes in this field.

You can also type specific path code libraries in this field. You can type up to 10 path codes at a time. For each path code, SETOWAUT modifies the library and associated IFS directory (specfiles path) accordingly.

- **Secure Log path:** A value of Y or N is valid for this field.
Use “N” to not secure JDE log paths. The recommended value is “N”.
Use “Y” only if there is a need to secure the log paths. One reason to secure JDE log paths may be if logs are being deleted without permission.
Only DB administrators have permission to access the logs in the log path.
- **Secure Objects for Coexistence:** Use this field to secure all (coexistence or non-coexistence) objects.
- ***ALLOBJ:** Use *ALLOBJ only if there is a need to secure all the objects contained in a library or directory. Plan carefully because *ALLOBJ can be

very time intensive. If there is a chance of a user getting past the library authority, it may be necessary to secure all the objects in the library.

Not only do the libraries and root directories have restrictions, but all the objects contained in the libraries or directories are restricted (*PUBLIC *EXCLUDE).

The LIBCRT attribute is set to *AUTL.

Datapath libraries and all the objects contained within have ownership of JDE.

- ***COEXIST:** Use *COEXIST if you have WorldSoftware installed on the AS/400 and it is sharing data paths.

Not only do the datapath libraries have restrictions, but all the objects contained in the datapath libraries are restricted (*PUBLIC *EXCLUDE).

The LIBCRT attribute is set to *AUTL.

The datapath libraries and all the objects contained within have ownership of JDE.

- ***NONCOEXIST:** Use *NONCOEXIST if you do not have WorldSoftware installed or if it is installed but not sharing any OneWorld libraries.

Only the libraries or root directories have restrictions placed on *PUBLIC.

Objects contained in any OneWorld library, with the exception of the System Library, allow *PUBLIC *ALL. The LIBCRT attribute is set to *ALL.

Datapath libraries have ownership of JDE.

Path code libraries or directories have ownership of ONEWORLD.

SETOWAUT authorization lists

If using types *FULL or *PROF, the SETOWAUT tool also creates the ONEWORLD and OWADMINL authorization lists (if they do not exist). In addition, SETOWAUT makes the following authority changes:

- Changes owner of both lists to ONEWORLD
- Adds JDE to both lists
- Changes *PUBLIC entry to *EXCLUDE in both lists

The ONEWORLD authorization list supports the following tasks:

- Secures all OneWorld libraries with the exception of the System Library (B7322SYS).
- Secures all database tables for *COEXIST.
- Secures all OneWorld objects (except System Library) for ALLOBJECTS(*ALLOBJ).

The OWADMINL authorization list is used in the following ways:

- Any user who performs basic ONEWORLD administration (Start, End, Clear IPC, etc.) on the AS/400 system must be added to this list (OWADMINL). Note that CRTOWADPRF is a supplied command that adds users to this list.
- Used to secure the System Library and all the objects contained within the System Library.

Verifying results for SETOWAUT (non-coexistence environment)

You should verify that your libraries, objects, user profiles, and IFS directories are secured the way you planned. The following list explains the tasks required for each:

- **Verifying libraries:** Use the following two commands:

```
-DSPOBJAUT OBJ (xxxxxx) OBJTYPE (*LIB)  
-DSPLIBD LIB (xxxxxx)
```

Here, *xxxxx* is the name of your data source libraries, pathcode libraries, or System Library.

Use the `DSPOBJAUT` command to verify that the library is secured by authorization lists properly (refer to Figure 220, Figure 222, and Figure 224). Use the `DSPLIBD` command to verify that the `*CRTAUT` parameter is set correctly to give default public authority for users to create new objects in the library (refer to Figure 221, Figure 223, and Figure 225).

The following six displays show examples of what you can expect to see for each of the OneWorld libraries. The highlighted information in the following illustrations should correspond to the information that appears on your screen.

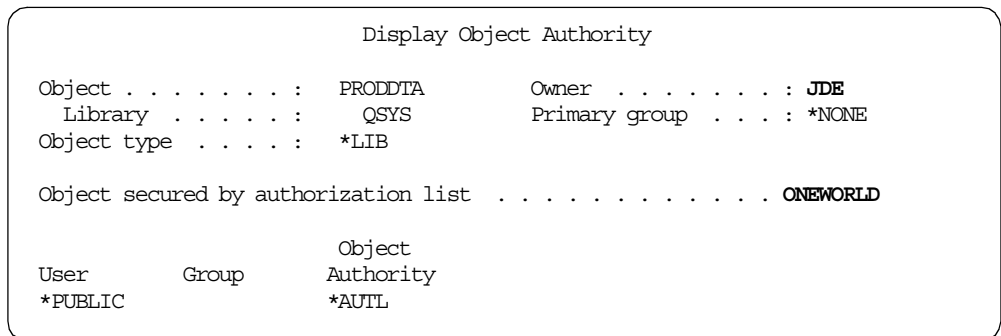


Figure 220. *DSPOBJAUT* data source

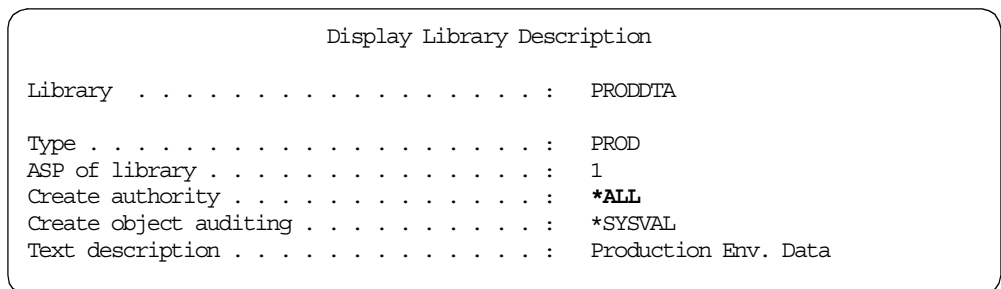


Figure 221. *DSPLIBD* data source

```

                                Display Object Authority
Object . . . . . : PRODB733      Owner . . . . . : ONEWORLD
  Library . . . . . : QSYS        Primary group . . . : *NONE
Object type . . . . . : *LIB

Object secured by authorization list . . . . . : ONEWORLD

User      Group      Object
*PUBLIC   Authority
          *AUTL

```

Figure 222. DSPOBJAUT path code

```

                                Display Library Description
Library . . . . . : PRODB733

Type . . . . . : PROD
ASP of library . . . . . : 1
Create authority . . . . . : *ALL
Create object auditing . . . . . : *SYSVAL
Text description . . . . . :

```

Figure 223. DSPLIBD path code

```

                                Display Object Authority
Object . . . . . : B7332SYS      Owner . . . . . : ONEWORLD
  Library . . . . . : QSYS        Primary group . . . : *NONE
Object type . . . . . : *LIB

Object secured by authorization list . . . . . : OWADMINL

User      Group      Object
*PUBLIC   Authority
          *AUTL

```

Figure 224. DSPOBJAUT B7332SYS

```

                                Display Library Description
Library . . . . . : B7332SYS

Type . . . . . : PROD
ASP of library . . . . . : 1
Create authority . . . . . : ONEWORLD
Create object auditing . . . . . : *SYSVAL
Text description . . . . . :

```

Figure 225. DSPLIBD B7332SYS

- **Verifying objects:** Objects within your System Library (B7332SYS) have their authorities changed by SETOWAUT. The authority changes for those objects

within System Libraries that either contain the attributes CLLE or CLP or those programs that share the same name as a command. You can view the authority on these objects using the following commands. The required parameters are object name, object type (*PGM or *CMD), and the name of the library where these objects reside:

```
DSPOBJAUT OBJ(yyyyy/xxxxxx) OBJTYPE(*PGM)
```

Here, *xxxxx* is the name of your object, *yyyyy* is the name of your library, and **PGM* or **CMD* is the object type.

```
DSPPGM PGM(B7332SYS/xxxxxx)
```

Here, *xxxxx* is the name of your program.

Use the DSPOBJAUT command to verify that the objects are secured by authorization lists properly (Figure 226). The highlighted information in the following two displays should correspond to the information that appears on your screen.

```

                                Display Object Authority
Object . . . . . : UNPAK           Owner . . . . . : ONEWORLD
Library . . . . . : B7332SYS      Primary group . . . . . : *NONE
Object type . . . . . : *PGM

Object secured by authorization list . . . . . : OWADMINL

User      Group      Object
*PUBLIC   *PUBLIC      *AUTL

```

Figure 226. DSPOBJAUT object

Use the DSPPGM command to verify that the user profile parameter is changed to *OWNER so that the program runs under both the current user's authority and the owner's authority. Also verify that the Use adopted authority parameter from previous levels is allowed (*YES) as in Figure 227.

```

                                Display Program Information
Program . . . . . : DB_ROUTER      Library . . . . . : B7332SYS
Owner . . . . . : ONEWORLD
Program attribute . . . : CLP

Program creation information:
Program creation date/time . . . . . : 02/07/00 09:52:21
Type of program . . . . . : OPM
Source file . . . . . : QCLSRC
Library . . . . . : OWB7332_D
Source member . . . . . : DB_ROUTER
Source file change date/time . . . . . : 02/07/00 07:29:05
Observable information . . . . . : *ALL
User profile . . . . . : *OWNER
Use adopted authority . . . . . : *YES
Log commands (CL program) . . . . . : *JOB
Allow RTVCLSRC (CL program) . . . . . : *YES
Fix decimal data . . . . . : *NO

```

Figure 227. DSPPGM program object

Authority for objects in data sources and path codes should remain the same after you run SETOWAUT. You can see this by displaying the authority for an object in each library before and after you run SETOWAUT.

```
DSPOBJAUT OBJ (yyyyy/xxxxxx) OBJTYPE (*PGM)
```

Here, *xxxxx* is the name of your object, *yyyyy* is the name of your library, and **FILE* is the object type.

Figure 228 shows an example of DSPOBJAUT. While your results may not look exactly the same as in the example, your before and after results of the highlighted fields should be identical.

```

                                Display Object Authority
Object . . . . . : F0002          Owner . . . . . : JDE
Library . . . . . :  PRODCIL      Primary group . . . . . : *NONE
Object type . . . . . : *FILE

Object secured by authorization list . . . . . : *NONE

User      Group      Object
JDE              Authority
*PUBLIC              *ALL
  
```

Figure 228. DSPOBJAUT object

- **Verifying user profiles:** After you run SETOWAUT, you can view the following user profiles and authorization lists to verify whether the information is correct. Compare the highlighted information in the following illustrations for accuracy.

Use the Display User Profile (DSPUSRPRF) command to check user profiles:

```
DSPUSRPRF xxxxx
```

Here, *xxxxx* is user profile ONEWORLD, JDE, your Security Profile, and your DB Administrator profile (see Figure 229 and Figure 230, and Figure 231 and Figure 232 on page 450).

```

                                Display User Profile - Basic
User profile . . . . . : ONEWORLD

Previous sign-on . . . . . : 02/01/00 10:34:59
Sign-on attempts not valid . . . . . : 0
Status . . . . . : *ENABLED
Date password last changed . . . . . : 05/17/99
Password expiration interval . . . . . : *NOMAX
Set password to expired . . . . . : *NO
User class . . . . . : *USER
Special authority . . . . . : *JOBCTL
Group profile . . . . . : *NONE
Owner . . . . . : *USRPRF
Group authority . . . . . : *NONE
Group authority type . . . . . : *PRIVATE
Supplemental groups . . . . . : *NONE
Assistance level . . . . . : *SYSVAL
Current library . . . . . : *CRIDFT
Initial program . . . . . : *NONE
  Library . . . . . :
Initial menu . . . . . : *SIGNOFF
  Library . . . . . :
Limit capabilities . . . . . : *NO

```

Figure 229. DSPUSRPRF ONEWORLD

```

User profile . . . . . : JDE

Previous sign-on . . . . . : 02/18/00 10:42:49
Sign-on attempts not valid . . . . . : 0
Status . . . . . : *ENABLED
Date password last changed . . . . . : 05/17/99
Password expiration interval . . . . . : *NOMAX
Set password to expired . . . . . : *NO
User class . . . . . : *USER
Special authority . . . . . : *JOBCTL
                               *SAVSYS
Group profile . . . . . : *NONE
Owner . . . . . : *USRPRF
Group authority . . . . . : *NONE
Group authority type . . . . . : *PRIVATE
Supplemental groups . . . . . : *NONE
Assistance level . . . . . : *SYSVAL
Current library . . . . . : *CRIDFT
Initial program . . . . . : *NONE
  Library . . . . . :
Initial menu . . . . . : MAIN
  Library . . . . . : *LIBL
Limit capabilities . . . . . : *NO

```

Figure 230. DSPUSRPRF JDE

```

User profile . . . . . : JDEOW

Previous sign-on . . . . . : 02/18/00 10:42:49
Sign-on attempts not valid . . . . . : 0
Status . . . . . : *ENABLED
Date password last changed . . . . . : 05/17/99
Password expiration interval . . . . . : *NOMAX
Set password to expired . . . . . : *NO
User class . . . . . : *USER
Special authority . . . . . : *NONE
Group profile . . . . . : *ONEWORLD
Owner . . . . . : *GRPPRF
Group authority . . . . . : *NONE
Group authority type . . . . . : *PRIVATE
Supplemental groups . . . . . : JDE
Assistance level . . . . . : *SYSVAL
Current library . . . . . : *CRIDFT
Initial program . . . . . : *NONE
  Library . . . . . :
Initial menu . . . . . : *SIGNOFF
  Library . . . . . :
Limit capabilities . . . . . : *NO

```

Figure 231. DSPUSRPRF JDEOW (Security profile)

```

User profile . . . . . : OWDBADM

Previous sign-on . . . . . : 02/18/00 10:42:49
Sign-on attempts not valid . . . . . : 0
Status . . . . . : *ENABLED
Date password last changed . . . . . : 05/17/99
Password expiration interval . . . . . : *SYSVAL
Set password to expired . . . . . : *NO
User class . . . . . : *PGMR
Special authority . . . . . : *NONE
Group profile . . . . . : ONEWORLD
Owner . . . . . : *GRPPRF
Group authority . . . . . : *NONE
Group authority type . . . . . : *PRIVATE
Supplemental groups . . . . . : JDE
Assistance level . . . . . : *SYSVAL
Current library . . . . . : *CRIDFT
Initial program . . . . . : *NONE
  Library . . . . . :
Initial menu . . . . . : MAIN
  Library . . . . . : *LIBL
Limit capabilities . . . . . : *NO

```

Figure 232. DSPUSRPRF OWDBADM (OneWorld DB Administrator)

- **Verifying authorization lists:** Use the command Display Authorization List (DSPAUTL) to view the authorization list authorities:

```
DSPAUTL xxxxxx
```

Here, xxxxx is ONEWORLD or OWADMINL. See Figure 233 and Figure 234 for a reference.

```

                                Display Authorization List
Object . . . . . : ONEWORLD      Owner . . . . . : ONEWORLD
Library . . . . . : QSYS         Primary group . . . : *NONE

      Object  List
User   Authority Mgt
ONEWORLD  *ALL    X
JDE      *ALL
*PUBLIC  *EXCLUDE

```

Figure 233. DSPAUTL ONEWORLD

```

                                Display Authorization List
Object . . . . . : OWADMINL     Owner . . . . . : ONEWORLD
Library . . . . . : QSYS         Primary group . . . : *NONE

      Object  List
User   Authority Mgt
ONEWORLD  *ALL    X
OWBADM    *USE
*PUBLIC  *EXCLUDE

```

Figure 234. DSPAUTL OWADMINL

- **Verifying IFS directories:** To view IFS directories (specification files), enter the Work With Links (WRKLNK) command on the command line, and page down until you see your specification file directory. Then select option **9** to work with authority. The highlighted items in Figure 235 show an example of verifying the security directory that holds your specification files.

```

                                Work with Authority
Object . . . . . : /PRODB733
Owner . . . . . : ONEWORLD
Primary group . . . . . : *NONE
Authorization list . . . . . : ONEWORLD

      Data      --Object Authorities--
Opt  User      Authority  Exist  Mgt  Alter  Ref
*PUBLIC      *AUTL
ONEWORLD     *RWX    X    X    X    X

```

Figure 235. IFS directory security settings

The security of your specification files should match highlighted items in Figure 237 on page 452 only if the SETOWAUT Secure Objects For Coexistence parameter was set to *ALLOBJ.

```

Work with Authority

Object . . . . . : /PRODB733/specfile/asvrctl.ddb
Owner . . . . . : ONEWORLD
Primary group . . . . . : *NONE
Authorization list . . . . . : ONEWORLD

      Data      --Object Authorities--
Opt  User      Authority Exist Mgt Alter Ref
-----
      *PUBLIC   *AUTL
      ONEWORLD  *RW      X    X    X    X

```

Figure 236. Specification file security settings

Important

For security reasons, the IFS directory with your specification files should be inaccessible through Windows Explorer.

15.2.6 Adding and removing additional administrators

After running SETOWAUT, you may need to add and remove additional administrators.

To add additional administrators, perform following steps.

1. On the command line, enter the following command:

```
CRTOWADPRF USRPRF (user1 user2 user3. . . user10)
```

The Setup OneWorld User Profile (CRTOWADPRF) form appears (Figure 237).

```

Setup OneWorld User Profile (CRTOWADPRF)

Type choices, press Enter

User Profile. . . . . _____ Name
      + for more values _____
Profile Type . . . . . *USER _____ *USER *ADMIN
Initial program to call . . . . *NONE _____ BV3C, *NONE, *SAME
LIBRARY NAME: . . . . . _____ JDEOW, library name, blank

```

Figure 237. CRTOWADPRF command

2. Complete the following fields and press Enter:

Note

The initial program to call is BV3C, by default, for OneWorld. This program sets the AS/400 system to provide a choice of environments at signon. A user with an administrator profile who signs on to an environment can then perform OneWorld commands on the AS/400 server.

a. OneWorld Admin Profile

You can add up to 10 administrators at a time.

b. Profile Type

Type *USER to grant the profiles with basic administration capabilities, such as STRNET, ENDNET, CLRIPC, SAW, CLRLCK, DSPIPC, DSPSTMF, IPCS, LINKBSFN, and PID2JOB.

Type *ADMIN if the profiles need the *USER administration capabilities as well as additional rights to PORTTEST and RUNUBE.

c. Initial program to call

Type BV3C if you want the system to display a list of environments when the administrators sign on to OneWorld, *SAME to use the current initial program setting, or *NONE to remove the initial program setting.

To remove additional administrators, enter the following command in the command line:

```
RMVOWADPRF USRPRF (user)
```

The Remove OW Profile Authority (RMVOWADPRF) form appears. See Figure 238.

Remove OW Profile Authority (RMVOWADPRF)

Type choices, press Enter.

User Profile _____ Name

Figure 238. RMVOWADPRF command

Chapter 16. Backup and recovery

This chapter covers the basic concepts of backup and recovery on an AS/400 system and some strategies specific to OneWorld.

16.1 AS/400 backup and recovery overview

Backup and recovery options for J.D. Edwards OneWorld on an AS/400 system are managed using *native AS/400 facilities*. Backup and Recovery Media Services/400 (BRMS/400) provides automated backup and archival operations, and tape management services. For additional information on BRMS/400, refer to *AS/400 Backup Recovery and Media Services*, SC41-4345.

An integral part of OS/400 provides many facilities to save and restore information on the AS/400 system. Depending on backup requirements and the operational environment, they can be selected from menu options, executed from a command line, or executed in user written programs.

While we do not develop a full backup and recovery strategy in this section, we discuss some considerations and include some example commands for a reference. This chapter does not cover all save and restore scenarios because each one is unique to their save strategy or reason for restore. Please refer to *Backup and Recovery*, SC41-5304, for more information on the subject.

Figure 239 on page 456 through Figure 242 on page 459 illustrate the functions performed by the standard AS/400 save and restore menu options and commands.

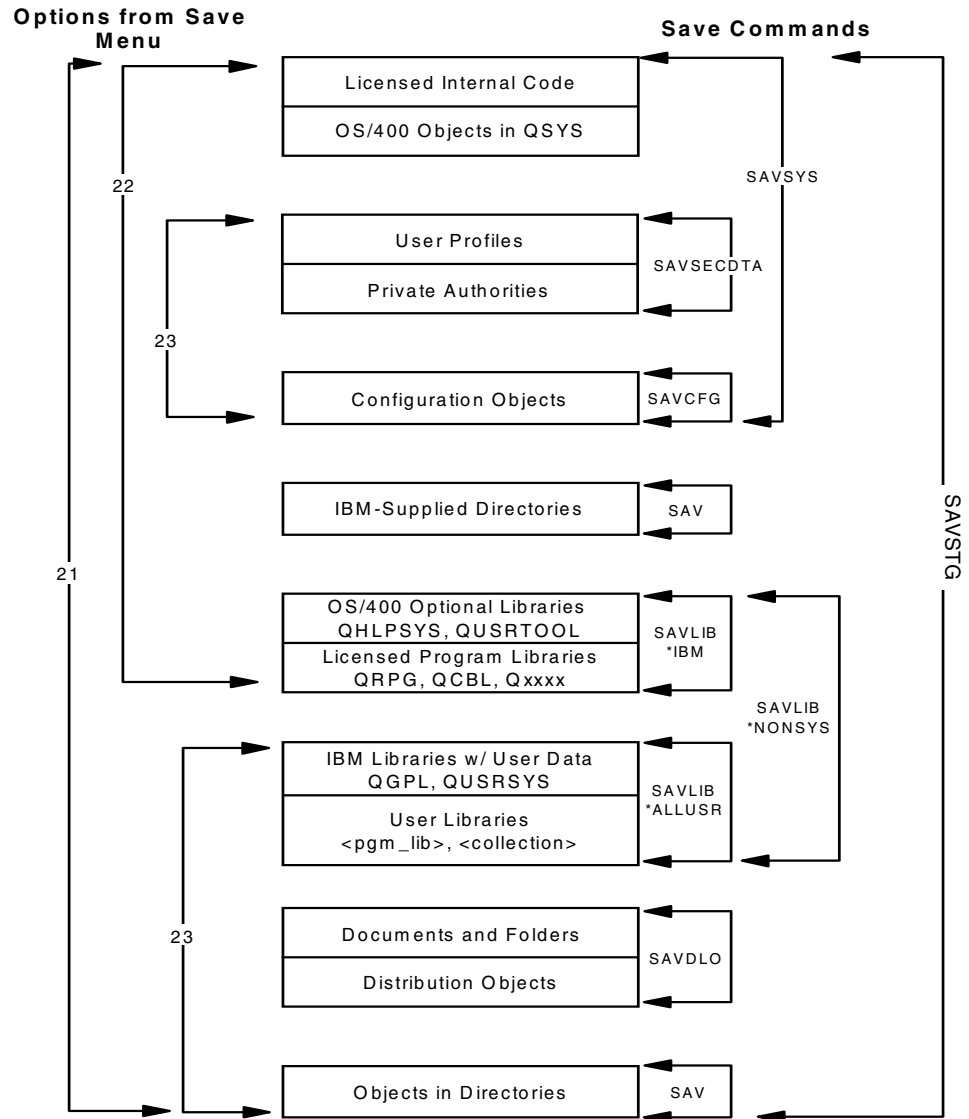


Figure 239. Library save commands and Save menu options

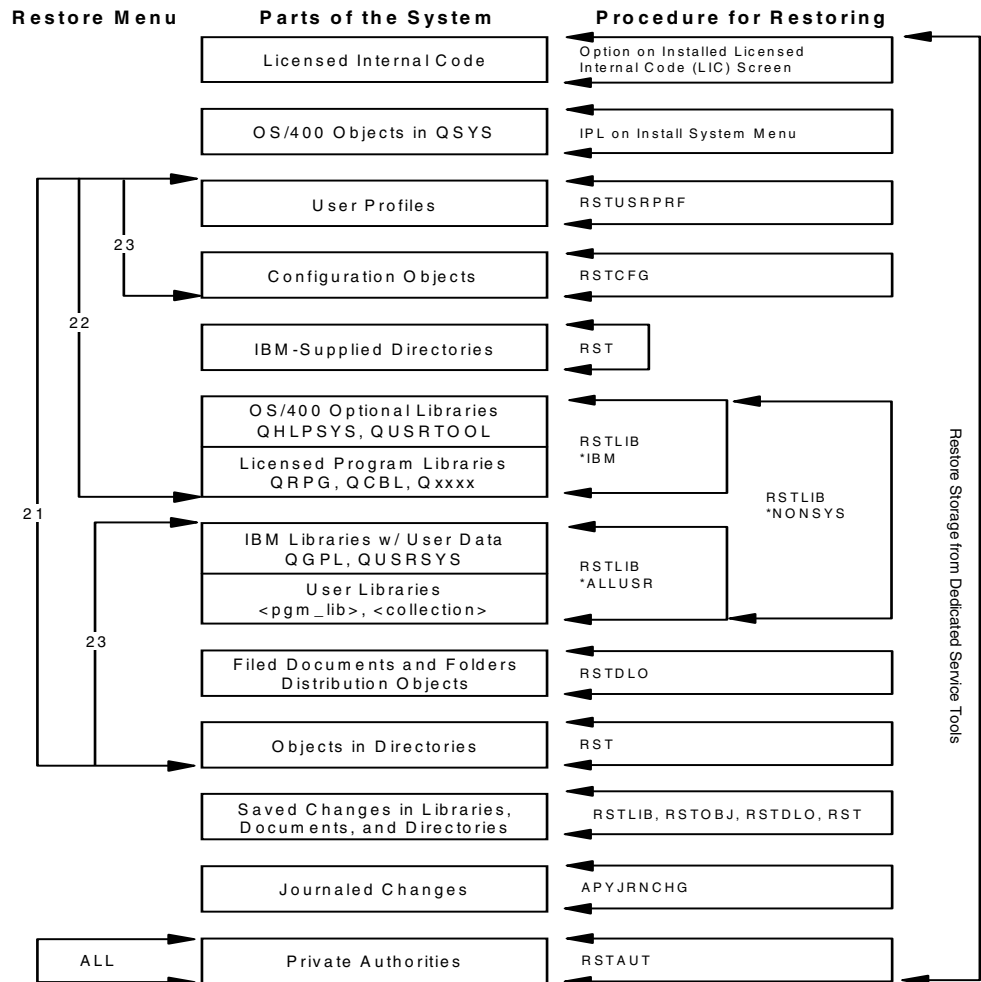


Figure 240. Library restore commands and Restore menu options

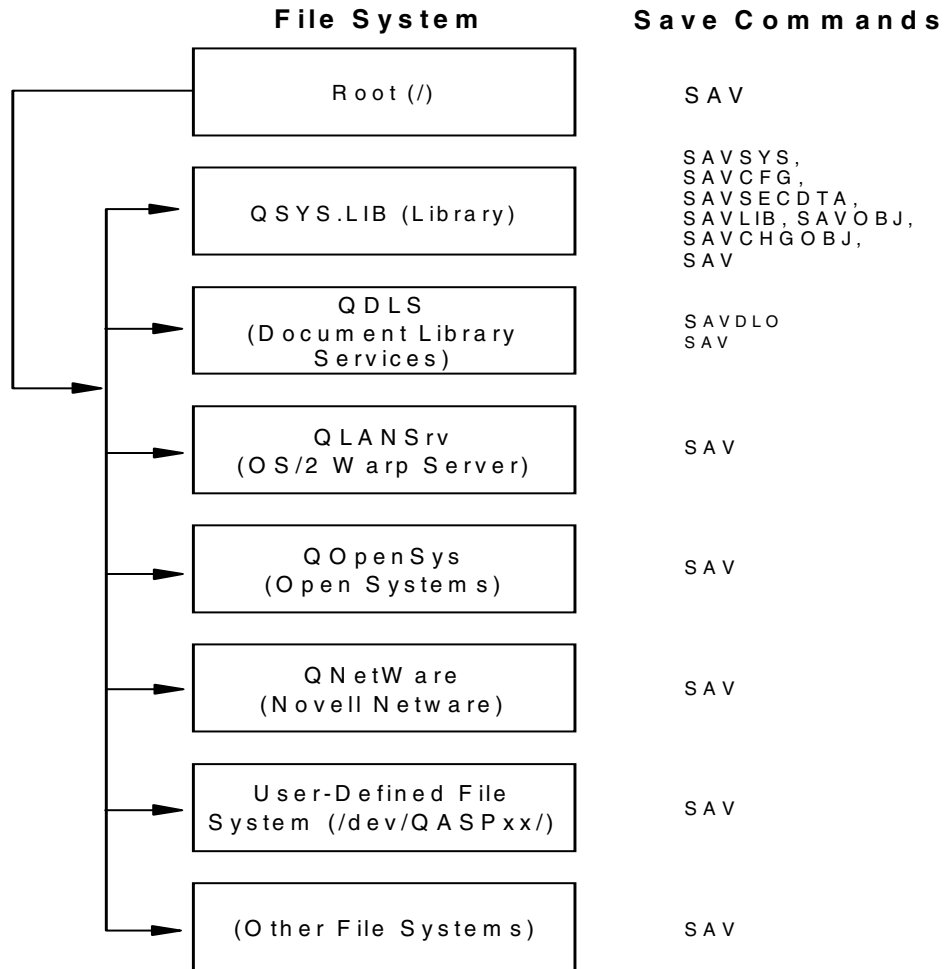


Figure 241. Save commands for the integrated file system (IFS)

Note: The following file systems cannot be saved:

- NFS
- QFileSvr.400
- QOPT
- QNTC

Restore Commands	File System	Save Commands
RST	Root (/)	SAV
SC41-5304 Chapters 12 & 13 RSTUSRPRF, RSTAUT, RSTCFG, RSTLIB, RSTOBJ, RST	QSYS.LIB (Library)	SAVSYS, SAVCFG, SAVSECDTA, SAVLIB, SAVOBJ, SAVCHGOBJ, SAV
RSTDLO RST	QDLS (Document Library Services)	SAVDLO SAV
RST	QLANSrv (OS/2 Warp Server)	SAV
RST	QOpenSys (Open Systems)	SAV
RST	QNetWare (Novell Netware)	SAV
RST	User-Defined File System (/dev/QASPxx/)	SAV

Figure 242. Save and restore commands for the integrated file system (IFS)

16.1.1 OS/400 save and restore commands

All AS/400 save and restore commands are provided in the base OS/400 operating system. If you need backup and save media management, AS/400 Backup and Recovery Media Services (BRMS) can accomplish this.

Save and restore functions can be accessed several different ways: menus via the GO command, indirectly through other commands, or the commands themselves.

The common Save and GO menu commands are:

- **SAVDLO**: Save Document Library Objects
- **SAVLIB**: Save Library
- **SAVOBJ**: Save Object
- **SAVSECDTA**: Save Security Data
- **SAVSTG**: Save Storage
- **SAVCFG**: Save Configuration
- **SAVSYS**: Save System
- **SAVCHGOBJ**: Save Changed Objects
- **SAV**: Save Integrated File System Object
- **GO CMDSAV**: Displays a menu of save commands
- **GO SAVE**: Displays a menu of save options
- **GO BACKUP**: Displays a menu of backup tasks

The common Restore and GO menu commands are:

- **RSTDLO**: Restore Document Library Objects
- **RSTLIB**: Restore Library
- **RSTOBJ**: Restore Object(s)
- **RSTAUT**: Restore User Authorities to Objects
- **RSTCFG**: Restore System Configuration
- **RSTUSRPRF**: Restore User Profiles and Authority Tables
- **RST**: Restore Integrated File System Object(s)
- **GO CMDRST**: Displays a menu or panel group of restore commands
- **GO RESTORE**: Displays a menu or panel group of restore tasks

Note that there is no restore system command since this is done as a dedicated service tools function. Nor is there is a restore changed object command. There are additional commands, but they are not listed or described here.

This chapter does not intend to explain each command in detail. If you want detailed information on the save and restore commands, refer to *Backup and Recovery*, SC41-5304, or *CL Reference*, SC41-5722.

16.1.1.1 Save commands requiring a dedicated system

A dedicated system (restricted state) on the AS/400 system only has the console and system jobs running. No users or other applications can use the system.

We list the commands that require a dedicated system. If you wrote your own save or backup programs using save-while-active and the system is in a dedicated mode, the save-while-active processing is ignored.

The commands are:

- SAVSYS
- SAVSTG
- SAVLIB LIB (*NONSYS)

16.1.1.2 Save commands not requiring a dedicated system

The following commands do not require a dedicated system (restricted state). However, use caution to prevent an interruption to production users due to the AS/400 system locking their objects during the execution of these saves. These save commands are:

- SAV
- SAVCFG
- SAVCHGOBJ OBJ (*ALL) LIB (<library-name>)
- SAVCHGOBJ OBJ (*ALL) LIB (*ALLUSR)
- SAVDLO
- SAVLIB LIB (*IBM)
- SAVLIB LIB (*ALLUSR)
- SAVOBJ OBJ (<object-name/library-name>)
- SAVSECDTA

16.1.2 Save-while-active

There is a capability on the AS/400 system that allows for saving information, while the environment is still available to users. This capability is called *save-while-active*. The save-while-active function of the AS/400 system is an option on several save commands. It allows you to use the system during all or

part of the backup process. It allows modification of objects during most of the time that they are being saved. In contrast, other Save functions provided on the system allow no access to the objects as they are being saved or only allow the objects to be read as they are being saved. The save-while-active function can be used with your other backup and recovery procedures to reduce or eliminate a system outage for particular save operations.

If you have a window of opportunity to perform a dedicated backup, we do not recommend this strategy. For detailed information on this capability, refer to *Backup and Recovery*, SC41-5304

16.1.3 Journaling

AS/400 system journaling uses two object types: *journals* (*JRN) and *journal receivers* (*JRNRCV). The journal acts like a funnel. All database table adds, changes, and deletes are received by the journal. The journal writes them to the journal receivers. The J.D. Edwards OneWorld product uses journaling and commitment control to complete a small set of its application processes. Because these processes only involve a portion of the OneWorld database, OneWorld journaling cannot be used for recovery purposes. For additional information on journals, journal receivers, and commitment control, refer to the *Backup and Recovery Guide*, SC41-5304.

16.1.4 ADSTAR Distributed Storage Manager

The IBM ADSTAR Distributed Storage Manager (ADSM) product 5769SV3 is an automated enterprise-wide client/server, policy-managed backup, and archive facility for network file servers and workstations. ADSM provides a centrally scheduled automated storage management solution. It can be used to backup or restore IFS and Windows NT server objects on the Integrated Netfinity Server (Integrated PC Server) down to the file level. This provides a suitable solution for performing system and data backups within OneWorld's distributed object, multi-platform architecture.

For more information, refer to the *ADSTAR Distributed Storage Manager for AS/400 Administrator's Guide*, GC35-0315.

16.1.5 Backup Recovery and Media Services (BRMS)

The IBM Backup Recovery and Media Services (BRMS) Licensed Program Product 5769-BR1 provides an AS/400 solution for policy-driven backup, recovery, hierarchical storage and media management. BRMS can be used for running and tracking routine operations involving tape media, ADSM Server storage, and related devices. It can run on a single AS/400 system or within a network of AS/400 systems supporting a shared media inventory. Media content information can be replicated on one or more systems in a network, enabling them to serve as data recovery centers. BRMS can be used in conjunction with the IBM AS/400 Job Scheduler (5769-JS1) to provide a very robust and flexible unattended automated backup strategy. A save strategy using multiple tape drives operating concurrently is recommended when performing unattended backups on systems with 300 or more gigabytes of disk. A 3494 Tape Library can further automate this approach if 3590 or 3490 tape drives are used.

For more information, refer to *Backup Recovery and Media Services for AS/400*, SC41-5345.

16.2 Backup and recovery requirements

The backup and recovery requirements depend on the availability goals for a specific installation. Consider the following factors:

- Cost to the business resulting from a loss of availability during the failure
- Probability of the failure occurring
- Cost of the backup strategy such as operator time, unavailability during backup, media cost, storage costs, and so on

The cost and benefits vary depending not only on the location of the installation, but also on the company policies. They also depend on the availability of the required skills to perform the backup and recovery operations.

The strategy should cover the loss of the database, as well as the suite of application code. The strategy should consider recovery from a disaster resulting in the loss of the computer site. It must include the following components:

- **System:** Including the operating system (OS/400), user profiles, authorities, configurations, system, and network values.
- **Application software:** Required for normal operations of the business, including compilers, utilities libraries, application and general purpose libraries (QGPL), IBM licensed program libraries, job descriptions, output queues, data areas, message queues, and other application dependent objects.
- **Databases:** Containing the organization's information.

If you have a large system or a very short window of time in which to perform a backup, the 3590 or 3570 tape drives are recommended. They are high speed, high capacity, and very reliable tape drives. The 3590 tape drive is recommended on systems with more than 100 GB of disk.

Beginning with V4R4M0, you can perform save operations while using more than one tape drive simultaneously. This parallel save approach can be used to further reduce backup time. However, there are some restrictions. For more information, refer to the *Backup and Recovery Guide*, SC41-5304.

16.3 Saving in a J.D. Edwards OneWorld environment

This section outlines a simple backup strategy that may satisfy the needs of a J.D. Edwards OneWorld installation. We categorize the J.D. Edwards OneWorld environment backup into three procedures:

- Initial OneWorld save and periodic entire system saves
- Weekly OneWorld save
- Daily OneWorld save

Tape media is not always reliable. Therefore, you may want to rotate your tape media and make additional copies of valuable data and keep those tapes off site. You may want to create multiple tape sets to protect your operation and create a unique and meaningful naming scheme for them.

16.3.1 OneWorld initial and periodic system saves

An initial save of the J.D. Edwards OneWorld environment is done after the initial installation is complete. Periodic entire system saves are recommended on a monthly basis, as well as before or after major system changes, such as:

- Hardware addition/removals
- Extensive system configuration changes
- Operating system fixes (PTFs)
- Application software additions, removals, and upgrades

For this example, we suggest that you perform a full system backup(entire system save). This can be accomplished by selecting option 21 from the Save menu.

Option 21 from the Save menu

This saves the entire system, including the AS/400 system library, user profiles, system configuration, IBM licensed program product libraries, all user libraries, document libraries, and the integrated file system (IFS).

Option 21 requires a dedicated system or restricted state, which means that no subsystems, except the controlling subsystem (for example QCTL), is available, and only to support the console. Upon completion, the controlling subsystem is restarted, and the system is made ready for production. If the STRNET command is not in the IPL start-up program, it has to be issued by someone such as the system operator to start the J.D. Edwards OneWorld subsystem (JDEB733).

Figure 243 shows an example of the display you see when you issue the GO SAVE command.

```
SAVE                               Save                               System: ASM23
Select one of the following:
Save Data
  1. Files
  2. Libraries
  3. Documents and folders
  4. Programs
  5. Other objects
  6. Changed objects only
  7. Licensed programs
  8. Security data
  9. Storage
 10. Configuration
 11. Objects in directories
More...
Selection or command
==> 21
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=Information Assistant
F16=AS/400 Main menu
(C) COPYRIGHT IBM CORP. 1980, 1998.
```

Figure 243. AS/400 save menu display

Select option **21** and press Enter to view the information shown in Figure 244 on page 464.

```

                    5769SS1 V4R3M0 980729
Display Device . . . . . : DSP01
User . . . . . : QSECOFR
                    Save the Entire System

What this option will do
o End all subsystems
o Save the Licensed Internal Code
o Save the operating system
o Save the security data
o Save the device configuration objects
o Save all user libraries (including libraries for licensed programs)
o Save all documents and folders
o Save all distribution and mail objects
o Save all directories
o Start the controlling subsystem
What this option will not do
o Save the contents of job queues, output queues, or data queues that
  exist on the system
Press Enter to continue.
F3=Exit  F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 1998.

```

Figure 244. First display after selecting option 21

When you press Enter to continue, the display shown in Figure 245 appears.

```

                    5769SS1 V4R3M0 980729
Display Device . . . . . : DSP01
User . . . . . : QSECOFR
                    Specify Command Defaults

Type choices, press Enter.
Tape devices . . . . . TAP01      Names
Prompt for commands . . . . . N      Y=Yes, N=No
Check for active files . . . . . N      Y=Yes, N=No
Message queue delivery . . . . . *NOTIFY  *BREAK, *NOTIFY
Start time . . . . . *CURRENT  *CURRENT, time
Vary off network servers . . . . . *ALL      *NONE, *ALL, *LANSERVER,
                                           *NETWARE, *BASE, *AIX,
                                           *WINDOWSNT

F3=Exit  F12=Cancel
More..

F3=Exit  F12=Cancel  F19=Left  F20=Right  F24=More keys
Bottom

```

Figure 245. Second display after selecting option 21

Fill in the tape device name and decide if you want the system to prompt you for additional responses or run without any intervention. We suggest that the Prompt for Commands parameter be set to a “N” so the backup goes unattended, except for tape mount messages.

Leaving the “Y” in the *Prompt for commands* field takes you through each command. It also allows you to specify the overrides for each save command to be executed. By leaving the default, the save does not execute unattended since user input is needed, so you must press the Enter key. Selecting the default “Y” for the *Check for active files*, the tape media is checked to ensure that data is expired before it is written over. If the data is not expired, a message is sent to

change the tape, or to write over the existing data. Specify `*NOTIFY` for the Message queue delivery field. If left at the default, messages break in and affect the completion of the save. By leaving the default of `*CURRENT` for the Start time field, the save starts immediately, unless a start time is specified. The start time must be entered in a 24-hour format.

Pay close attention to the start time parameter if the save is not going to be performed immediately. If you have an Integrated Netfinity Server, enter `*ALL` for the Vary off network servers parameter. If you page down, you can leave the defaults, unless you are using a User Defined File System (UDFS). Specify "Y" for print system information for a report to be used in case of system recovery. Once you press the Enter key, the backup starts.

This backup or save, when completed, produces an offline copy of the entire system to tape. These tapes can be used for recovery purposes of the entire system or individual objects.

If you purchase the Application Development Toolset Licensed Program Product (LPP), you can use the AS/400 Programming Development Manager (PDM) or Source Entry Utility (SEU) to develop your own program by entering CL commands to run this save. You should be familiar with the OS/400 editor, CL commands, and how to create a program object.

16.3.2 Weekly OneWorld saves

A weekly OneWorld backup or save is the same as the initial save, except you do not have to save the OS/400 system library or the IBM licensed program libraries. This type of backup is referred to as a `*NONSYS`. For this save scenario, we assume that only the J.D. Edwards OneWorld environment needs saving and the IBM libraries are saved when needed.

Insert a blank tape into the tape drive. Use the `INZTAP` command to initialize the tape. If your media density does not comply with your tape drive, you may have to specify a different density. Your display should appear as shown in Figure 246 on page 466.

```

                                Initialize Tape (INZTAP)

Type choices, press Enter.

Device . . . . . > TAP01      Name
New volume identifier . . . . . > WKSAV1    Character value, *NONE...
New owner identifier . . . . . *BLANK
Volume identifier . . . . . *MOUNTED      Character value, *MOUNTED
Check for active files . . . . . > *NO      *YES, *NO, *FIRST
Tape density . . . . . *DEVTYPE          *DEVTYPE, *CTGTTYPE, *QIC120...
Code . . . . . *EBCDIC                *EBCDIC, *ASCII
End of tape option . . . . . *REWIND      *REWIND, *UNLOAD
Clear . . . . . *NO                    *NO, *YES

                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 246. Initialize Tape (INZTAP) command

The weekly save should, at the very least, consist of two separate saves. A *NONSYS save includes all non-system related libraries, using the SAVLIB LIB(*NONSYS) command. This command must be run while the system is in a restricted state.

The second type of save backs up the IFS directories, using the SAV command. The SAV command is structured like this:

```

SAV DEV('QSYS.LIB/TAP01.DEVD') OBJ('/<path_code_1>') ENDOPT(*LEAVE)
SAV DEV('QSYS.LIB/TAP01.DEVD') OBJ('/<path_code_2>') ENDOPT(*LEAVE)
SAV DEV('QSYS.LIB/TAP01.DEVD') OBJ('/<path_code_n>') ENDOPT(*REWIND)

```

Here, *path_code_n* is equal to the path codes installed on your system (for example, PRODB733 or CRPB733). Figure 247 shows an example of the SAV display when you press the F4 key to prompt it.

```

                                Save Object (SAV)

Type choices, press Enter.

Device . . . . . > 'QSYS.LIB/TAP01.DEVD'

                                + for more values

Objects:
  Name . . . . . > '/PRODB733'

  Include or omit . . . . . *INCLUDE      *INCLUDE, *OMIT
                                + for more values
  Directory subtree . . . . . *ALL        *ALL, *DIR, *NONE, *OBJ
  Save active . . . . . *NO            *NO, *YES, *SYNC
  Output . . . . . *NONE

  Volume identifier . . . . . *MOUNTED
                                + for more values
  Label . . . . . *GEN

                                                                More...
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 247. The SAV command display

We also recommend that you backup the Deployment Server weekly if possible. You must shut down the database before you create any backups. Refer to Chapter 6 in the J.D. Edwards *OneWorld Server and Workstation Administration Guide* for assistance. For more information, you can also refer to *Backup and Recovery*, SC41-5304, or *The System Administrator's Companion to AS/400 Availability and Recovery*, SG24-2161.

16.3.3 Daily OneWorld saves

A daily OneWorld backup or save can be exactly like the J.D. Edwards OneWorld weekly backup. However, as with most commercial installations, system availability must be kept to a maximum. There are two distinct functions that can assist in this process. One function is to use the SAVCHGOBJ command on the J.D. Edwards OneWorld Enterprise Server database libraries. This saves all objects that changed since the last full save (Reference Date = *SAVLIB). This can save time, since this backup does not save information that did not change since the last weekly OneWorld backup. Figure 248 on page 468 shows an example.

```

                                Save Changed Objects (SAVCHGOBJ)

Type choices, press Enter.

Objects . . . . . > *ALL          Name, generic*, *ALL
      + for more values
Library . . . . . > *ALLUSR       Name, generic*, *ALLUSR
      + for more values
Device . . . . . > TAP01         Name, *SAVF
      + for more values
Object types . . . . . *ALL       *ALL, *ALRTBL, *BNDDIR...
      + for more values
Journalled objects . . . . . > *YES   *NO, *YES
Reference date . . . . . *SAVLIB   Date, *SAVLIB
Reference time . . . . . *NONE     Time, *NONE
Volume identifier . . . . . *MOUNTED
      + for more values
Sequence number . . . . . *END     1-16777215, *END
Label . . . . . *LIB
File expiration date . . . . . *PERM   Date, *PERM
More...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 248. Save Changed Objects display

If you feel comfortable, you can select individual libraries on the *Library* field to save additional time. Another option (although not recommended) is to specify a reference date instead of *SAVLIB. Using this strategy, you can perform incremental backups daily, instead of a cumulative backup from the last full save. However, this complicates the restore process. Select *YES for the journaled objects to ensure any files that are only being journaled for commitment control (not High Availability) are also being backed up.

The J.D. Edwards OneWorld directories can be backed up using the SAV command. This command has a similar option through the parameter called CHGPERIOD. To use this option, you specify the reserved value *LASTSAVE. For the CHGPERIOD(*LASTSAVE) option to be effective, the initial and weekly J.D. Edwards OneWorld SAV commands must specify UPDHST(*YES). The default on the SAV command is UPDHST(*NO).

16.4 Restoring in a OneWorld environment

There are two main AS/400 restore scenarios. One scenario restores the entire system, including the system library due to an entire system or site failure. The other scenario restores the J.D. Edwards OneWorld environment.

16.4.1 Restoring the entire system

This section does not explain the entire system restore scenario because it is lengthy and documented in the manual *Backup and Recovery*, SC41-5304. However, this section lists the major steps and commands that are needed to accomplish this:

1. Restore or install System Licensed Internal Code from Alternate IPL device (CD-ROM or tape).

2. Install the operating system from the Alternate IPL device.
3. Restore User Profiles (RSTUSRPRF).
4. Restore Configuration (RSTCFG).
5. Restore All Libraries (RSTLIB).
6. Restore Document Library Objects (RSTDLO).
7. Restore the integrated file system (RST).
8. Apply Journal Changes (APYJRNCHG).
9. Restore public and private authorization (RSTAUT).
10. Validate the J.D. Edwards OneWorld environment (testing).
11. Start the OneWorld environment and make the system available to users.

16.4.1.1 Correct order for restoring objects associated with a journal

For the system to automatically re-establish your journaling environment, restore objects in this sequence:

1. Journals
2. Based-on physical files
3. Dependent logical files
4. Journal receivers

When these objects are in the same library, the system restores them in the correct sequence. When these objects are in different libraries, you must restore them in the correct sequence, or you must manually re-establish your journaling environment after the restore operation.

16.4.2 Restoring the OneWorld environment

If the entire system is not lost, you may just need to restore an individual library or file that was inadvertently deleted or has become damaged or corrupted in some way. This can be accomplished by using the RSTLIB, RSTOBJ, or the RST command for IFS objects. Sometimes it is easier to delete the object that you want to replace before you overlay it. Here is an example of the command you can use to restore the security workbench table:

```
RSTOBJ OBJ(F00950) DEV(TAP01) SAVLIB(SYSB733) OBJTYPE(*FILE) MBOPT(*ALL)
```

Since it is impossible to predict the different scenarios, we recommend that you refer to the *AS/400 Backup and Recovery*, SC41-5304.

16.4.3 System Managed Access Path Protection

System Managed Access Path Protection (SMAPP) can greatly reduce the amount of time it takes to perform an IPL after an abnormal end. It is an automatic function that runs without attention. SMAPP determines which access paths to protect without any intervention by the user. It automatically adjusts or makes changes in the environment such as when adding new applications or new hardware.

SMAPP does not require any setup, and you do not have to change your applications. You do not have to journal any physical files or even use journaling at all. You simply need to determine your policy for access path recovery. This should include evaluating:

- How long you can afford to spend rebuilding access paths during an IPL after a failure
- How to balance access path protection with other demands on system resources
- Whether to have different target times for recovering access paths for different ASPs

You may need to experiment with different target recovery times for access paths to achieve the correct balance for your system. If you configure additional user ASPs, you should also evaluate your access path recovery times.

SMAPP creates and manages its own internal journals and journal receivers. You cannot use these journals and journal receivers for any other journaling functions. They do not appear on the Work with Journals display. You do not need to, nor can you, save them to tape.

SMAPP requires additional auxiliary storage for journal receivers. However, SMAPP is designed to keep the additional disk usage to a minimum. SMAPP manages journal receivers and removes them from the system as soon as they are no longer needed.

Under SMAPP, the system looks at all access paths to determine how it can meet the specified target times for recovering access paths. It may not choose to protect an access path that you consider critical. When the system determines how to meet the target times for recovering access paths, it considers only access paths that are not explicitly journaled.

You can allow the system to determine which access paths to protect. Specify target recovery times for access paths for the entire system or for ASPs. Your system has a default recovery time for access paths for the entire system of 150 minutes when it is shipped. You can use the Edit Recovery for Access Paths (EDTRCYAP) command to see and change the target recovery times for access paths. This command also allows you to see how much space the system is using for System Managed Access Path Protection (SMAPP).

SMAPP provides a simple method to reduce your recovery time after an abnormal system end. SMAPP manages the required environment for you. You do not need to use any type of journal management to use SMAPP.

Refer to the manual *Backup and Recovery*, SC41-5304, for additional information on SMAPP.

Chapter 17. National language support

National language support (NLS) is a set of common standards that allows you to enter, display, store, retrieve, and print data in multiple languages, in different databases, and on different platforms.

This chapter provides information on the national language support topics you need to understand when using OneWorld on the AS/400 system. This chapter will be most interesting for those of you who are implementing OneWorld on the AS/400 system in a multilingual or non-English speaking environment.

17.1 AS/400 national language support overview

This section gives you an overview of the AS/400 system implementation of national language support (NLS).

17.1.1 Primary and secondary national language version

Each AS/400 system has one primary national language version (NLV). The primary national language version consists of code and textual data for each licensed program. The textual data (also called *machine readable information* (MRI)) has been customized by IBM to meet the requirements of the language feature. It may be fully, partly, or not translated into the language. Cultural system values are initialized for the primary language as shown in Figure 249.

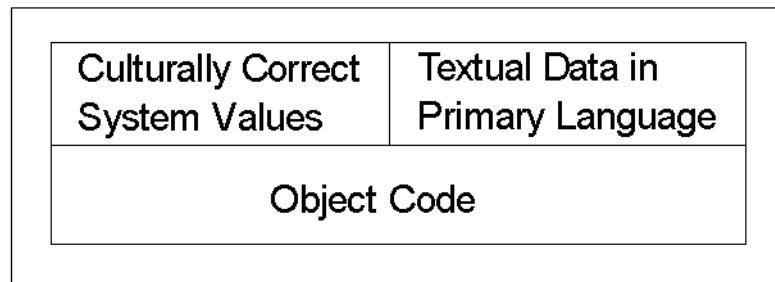


Figure 249. OS/400 primary language national language version (NLV)

Each AS/400 system must always have a “primary language”. This is the first language installed on the system and is the language that is used for servicing purposes. For example, the system history log contains entries in the primary language (assuming that these entries have been translated).

Appendix C, “National language versions on AS/400 systems” on page 817, shows the available national language versions. If a national language version is not available for a specific language, one of the available national language versions must be used. In this case, customers have to set the culturally correct system values themselves.

Any languages other than the primary language are, by definition, secondary languages. Installing a secondary language just requires the MRI in the appropriate language. Software ordered from IBM as a secondary language cannot be used as a primary language since the program code is not contained within it.

A secondary language consists of textual data only for each licensed program. The distribution media contains textual data for all licensed programs.

When installing a secondary language, new containers for the textual data are created as shown in Figure 250. For library objects, libraries are named QSYS29XX, where XX is from the last two digits of the secondary national language feature code. For objects in directories, sub-directories named MRI29XX are created and only the textual data for the licensed programs currently installed on the system are copied from the distribution media to the containers.

If new licensed programs are added after initial installation of a secondary language, the secondary language must be installed again to load the secondary language part for the new licensed programs. Figure 250 shows how secondary language fits with primary language.

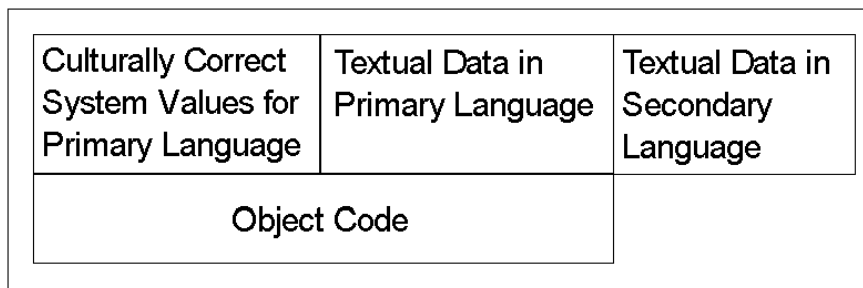


Figure 250. OS/400 secondary language NLV

Secondary languages require approximately 300 MB to 500 MB of disk space.

Multiple secondary languages can be installed on one AS/400 system. The maximum number of secondary languages on one AS/400 system depends on the available amount of disk space on the system and the availability of:

- A specific secondary language from the regional software distribution center
- Hardware (displays, keyboards, printers) for specific secondary languages

17.1.2 Language sensitive settings

Too often, national language support is mistakenly viewed as the task of simply translating the textual data part of a product into other national languages. The proper support is to provide facilities for the end user to obtain results that are culturally acceptable. The AS/400 system provides these facilities through cultural system values and job attribute values.

17.1.2.1 Cultural system values

The AS/400 system provides the default setting for the cultural information through system values. Table 36 shows the different system values used as defaults.

Table 36. Cultural system values

System value	Description
QCCSID	Coded character set identifier
QCHRID	Character set/code page

System value	Description
QCNTYID	Country identifier
QCURSYM	Currency symbol
QDATFMT	Date format
QDATSEP	Date separator
QDECFMT	Decimal format
QKBDTYPE	Character set/code page for keyboard
QLANGID	Language identifier
QLEAPADJ	Leap year adjustment
QSRTSEQ	Sort sequence
QTIMSEP	Time separator

Use the Display System Value (`DSPSYSVAL`) command to display a system value and the Work with System Value (`WRKSYSVAL`) command to display or change system values.

Important

Do not change the cultural system values on your AS/400 system unless you are fully aware of the consequences. By changing some system values (especially QCCID and QLANGID), you may impact your data.

17.1.2.2 Cultural job attribute values

Job attributes are set for a job when you sign on or when a job starts. Job attributes are set from the user profile associated with the job. The user profile often points to the default system values. The user profile may specify its own attributes for the cultural values, which allows multiple users with different needs to operate on the same AS/400 system. Table 37 specifies the cultural job attribute values.

Table 37. Cultural attribute values

Job attribute	Description
CCSID	Coded character set identifier
DFTCCSID	Default coded character set identifier
CNTYID	Country ID
DATFMT	Date format
DATSEP	Date separator
LANGID	Language ID
SRTSEQ	Sort sequence
TIMSEP	Time separator

Once initialized, the job attributes for that job can be changed.

The default CCSID cannot be changed directly since this value is set through a dependency of CCSID and LANGID, which is explained in 17.1.4.2, “The default CCSID” on page 476.

Important

Do not change the cultural values for job attributes or any values in user profiles on your AS/400 system unless you are fully aware of the consequences. By changing some cultural values (especially CCSID and LANGID), you may impact your data.

17.1.3 NLV setting for cultural values

Message ID CPX8416 in message file QCPFMSG is used during the installation to set cultural system values for the primary language.

For the primary language, message file QCPFMSG is located in the QSYS library. The message description can be displayed by entering the `DSPMSGD CPX8416` command. You see a screen like the example in Figure 251.

```

                                Display Formatted Message Text
                                System:  AS23
Message ID . . . . . :  CPX8416
Message file . . . . . :  QCPFMSG
  Library . . . . . :  QSYS

Message . . . . . :

QCHRID      697 37                QCURSYM    $ QDATFMT MDY QDATSEP /
QDECFMT     QLEAPADJ  0 QCCSID    37 QTIMSEP : QLANGID  ENU
QCNTRYID    US QIGCCDEFNT *NONE

Press Enter to continue.

F3=Exit  F11=Display unformatted message text  F12=Cancel

```

Figure 251. Display Formatted Message Text

Table 38 shows an example of CPX8416 in the U.S. English national language version.

Table 38. Message description CPX8416

Identifier	Value	Description
QCHRID	697 37	Character set/code
QCURSYM	\$	Currency symbol
QDATFMT	MDY	Date format
QDATSEP	/	Date separator
QDECFMT	(blank)	Decimal format
QLEAPADJ	0	Leap year adjustment

Identifier	Value	Description
QCCSID	37	Preferred NLV CCSID Note: QCCSID is initially set to 65535
QTIMSEP	:	Time separator
QLANGID	ENU	Language ID
QCNTYID	U.S.	Country ID

For a secondary language, message file CPFMSG is located in the QSYS29XX library. The message description can be displayed by entering the command:

```
DSPMSGD CPX8416 MSGF(QSYS29XX/QCPFMSG)
```

If there are different language users on a single system, it is possible to use message CPX8416 to set the cultural job values during sign-on for a user. An initial program specified in the user profile can add QSYS29XX to the system library list in front of QSYS, retrieve the values from message ID CPX8416, and issue the CHGJOB command using the retrieved cultural values.

Note

Changes made to the cultural system values do not change the contents of message CPX8416 in message file QCPFMSG in QSYS, and changes made to CPX8416 do not influence system values. For customers not using their intended national language version, or languages not having a national language version, the system values and content of CPX8416 should be kept synchronized except for system value QCCSID. You may not want to change system value QCCSID until you are sure that there is no impact to your data.

Similar to message CPX8416, licensed programs use messages to define their values for cultural processing.

17.1.4 Multilingual and multi-system database support

With the integrated database, the AS/400 system is an excellent Database Server. It can support both single language, multi-platform environments, and multilingual user environments. The system needs to know about the content of the database files to perform necessary conversions requested by applications and clients.

Just as the fields are described with column heading and text, the file description contains information about the actual encoding of the database records. This is the coded character set identifier (CCSID).

17.1.4.1 The importance of CCSID

Whenever a new, externally described database file is created, the AS/400 system stores information about the encoding (the CCSID) in the file description. Generally, all character fields in one database file have the same CCSID, although the AS/400 system allows the CCSID to be as granular as the field or column level.

All AS/400 systems are shipped with system value QCCSID 65535, which means no conversion. The reason for selecting this value is to stay compatible with previous releases and not to cause any impact on customer data if the value was set to the recommended setting for the national language version. Refer to Chapter 10 in *Speak the Right Language with Your AS/400 System*, SG24-2154, for more information.

The problem with 65535 is that it does not work when character data processing is performed with different code pages involved. For example, Distributed Relational Database Architecture (DRDA) and Client Access Express use CCSIDs to determine how to convert character data to the client system. If the CCSID is 65535, no conversion is performed and the data will appear as garbage to the PC client.

17.1.4.2 The default CCSID

When a job is started on the AS/400 system, the job attributes are initialized. In this section, we discuss the items that relate to CCSID issues and use Figure 252 for illustration purposes. For detailed information on all job attributes, refer to *OS/400 Work Management Guide*, SC41-5306.

The job attributes for the language ID and CCSID are taken from the user profile that starts the job. The user profile can define the values or point to the system values.

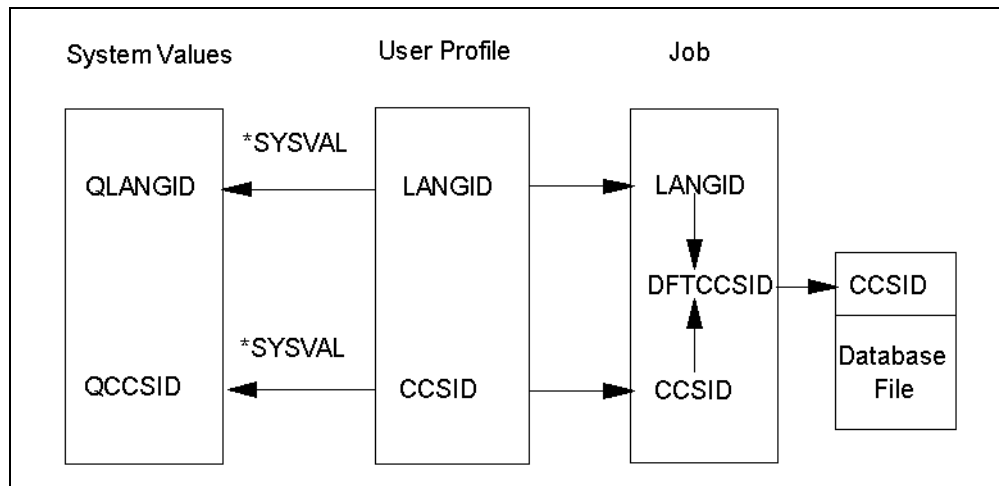


Figure 252. Default CCSID

The default job-level CCSID is determined as follows:

- If the job CCSID is 65535, the default CCSID is the CCSID corresponding to the language ID of the job.
- If the job CCSID is not 65535, the default CCSID is the same as the job CCSID.

Since the default for all user profiles is to use the system values, customers using a primary language without defining the correct QLANGID, may find that their database files are not tagged correctly.

For example, a customer in Israel uses uppercase and lowercase English (Feature Code 2924) as their primary language, but the keyboard used is in

Hebrew. Looking at the Table 62 on page 817, the language ID for feature 2924 is ENU and the associated CCSID is 37. Since the system value QCCSID is 65535, the default CCSID on this Israeli system is 37.

Any new externally described file (including the source file) that is not explicitly defined with a CCSID is created with the default CCSID.

For the Israeli customer, it means that the files carry the incorrect tag of CCSID 37. The language ID for Hebrew is HEB and the associated CCSID is 424, which is the correct CCSID tag for database files created with Hebrew keyboards.

As long as the data is used only by interactive applications on the AS/400 system, the data appears correctly. However, if a user wants to access the data with Client Access/400, the Hebrew characters do not appear correctly.

Note

Conversions within an AS/400 job can only occur when the CCSID of the job is different from the CCSID of the file and none of them are 65535.

Important

Do not change system or user profile values referring to the language ID or CCSID on your AS/400 system at this time. If your database files are incorrectly tagged, you need to plan for this change to your incorrectly tagged database files and system value. You must ensure that you change your current, incorrectly tagged database files and the system values as one task to avoid further inconsistencies.

17.1.4.3 Changing CCSIDs of physical files (tables)

You can change the CCSID of a physical file by using the Change Physical File (CHGPF) command. When the CCSID of a physical file is changed, its associated logical files are also changed to the new CCSID without having to delete and recreate the logical files.

There are a few restrictions that may prevent you from changing the CCSID of a physical file:

- A logical file built over the physical file has a sort sequence table, and the CCSID of the sort sequence table is incompatible with the new CCSID in the physical file.
- A select/omit, join logical file, or both perform select/omits or joins between physical file fields that have different CCSIDs.
- A join logical file with a sort sequence table where the CCSID of the logical file's secondary access path is different than the new CCSID for the physical file.
- CCSIDs are explicitly specified.
- A physical file has a physical file constraint. You must remove the physical file constraint before changing the CCSID of the physical file. When the CCSID is changed, the physical file constraint can be added again.

Changing the CCSID of a physical file does not change the content of the file, just the CCSID itself. The CCSID of the file should match the keyboard that entered the data into the file.

Note: When changing the CCSID of a source file, the last change date and time is updated. Refer to Chapter 10 in *Speak the Right Language with Your AS/400 System*, SG24-2154, for more information.

17.1.4.4 CCSID special values

Some of the values for the CCSID that have special meaning are:

- **65535:** An object having this CCSID specified does not participate in any conversion. This is also known as *HEX.
- **65534:** Informs the user of the object to look at a lower level for the actual CCSID. For example, the DSPFD shows -1, which instructs you to look at the DSPFFD to find the CCSID tagging each field. Not all fields have the same CCSID.
- **0:** Informs the user of the object to look at a higher level in the hierarchy for the actual CCSID used. For example, the file field description internally specifies 0 in the CCSID when all fields within the file have the same CCSID, informing the system to look in the file description. Externally, the DSPFFD command propagates the CCSID of the file description to each field description.

17.1.5 System CCSID support

The AS/400 system includes CCSID support in object types other than database files and jobs. For detailed information about CCSIDs for other object types, refer to *National Language Support*, SC41-5101, and *International Application Development*, SC41-5603. Figure 253 shows all AS/400 objects that support CCSIDs and how they relate to each other.

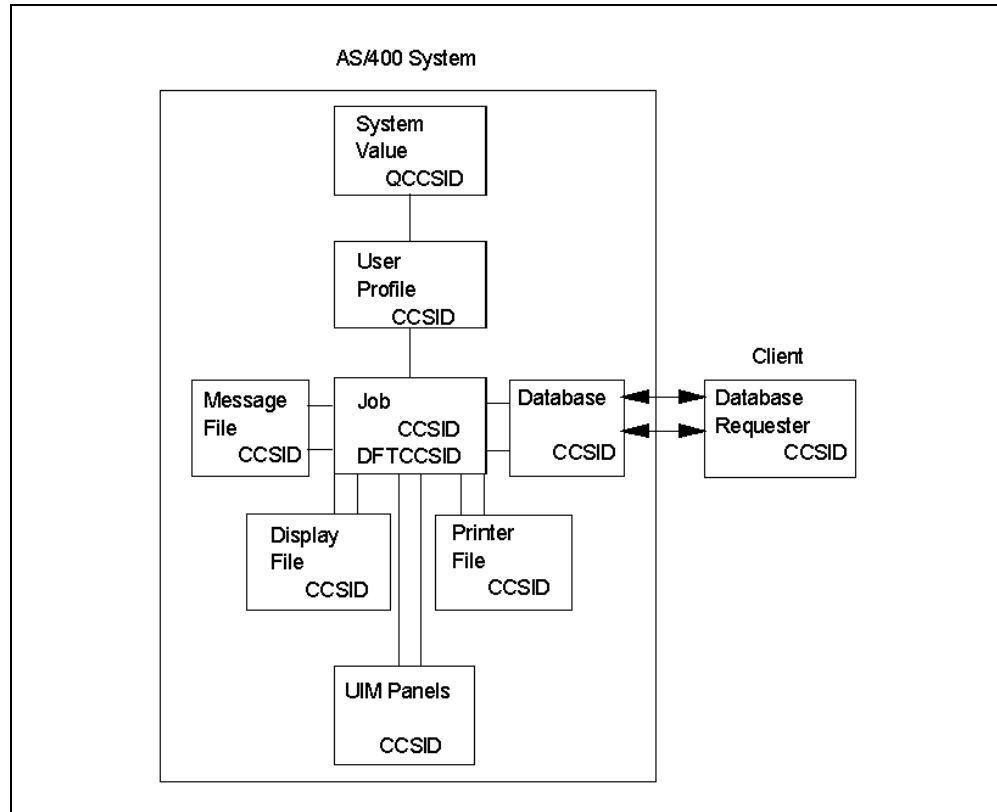


Figure 253. AS/400 objects that support CCSID

Remember these key points:

- The CCSID is just a tag in the object, an identifier to the outside world about the encoding of character data.
- The job attributes for CCSID and default CCSID are initiated by user profile values for CCSID and language ID, where the user profile may point to system values QCCSID and QLANGID.
- Database files (including source files) are created with the default CCSID in the file description unless explicitly defined.
- Automatic conversion of character data is performed between two objects when the CCSIDs of the involved objects are different and none of them is 65535.

17.2 OneWorld national language considerations

This section examines OneWorld national language support issues on the AS/400 system. From the national language support point of view, J.D. Edwards categorizes languages into three tiers: tier 1, tier 2, and tier 3.

Tier 1 languages are languages supported by the J.D. Edwards corporate translation and development departments for both software and documentation. The eight tier 1 languages of J.D. Edwards software are:

- Chinese Simplified
- English

- French
- German
- Italian
- Japanese
- Portuguese
- Spanish

Tier 2 languages are languages supported by in-house translation of software only. The seven tier 2 translations are:

- Chinese Traditional
- Danish
- Dutch
- Finnish
- Korean
- Norwegian
- Swedish

Tier 3 languages are languages supported through software translation by J. D. Edwards Business Partners only. The seven tier 3 translations are:

- Arabic (WorldSoftware only)
- Czech
- Greek (WorldSoftware only)
- Hungarian
- Polish
- Russian
- Turkish (Under consideration)

17.2.1 Overview

The language installation process copies the text from a language database to your production OneWorld database and merges the alternate language text with the original English base. You can choose to install one additional language during the initial OneWorld installation process or do a language-only plan (to install one or more languages) after OneWorld has been installed.

17.2.1.1 Language architecture

OneWorld's language architecture incorporates multinational language functionality for international customers. The software lets you specify your language preference for forms and reports. For example, users that share the same environment may want to view the same text in different languages, such as French, German, and English. All language text is stored in a central location and deployed to the individual workstation.

J.D. Edwards currently ships all software with a base language of English. The package build process allows you to build packages for multiple languages. Language text is accessed by the language preference code settings in the user profile associated with the alternate language installed.

Base language

OneWorld language support works in conjunction with the English base language. The base release contains English and must be installed before creating custom modifications to include changes or additions to the translated text. All control files must also have an English record prior to adding an alternate

language text record. Preferably, your AS/400 primary language should also be English.

Alternate language

The following two major components are included with an alternate language:

- Standard packages, which are installed on the Deployment Server, include both English and the alternate language installed. These language packages, which contain the desired language, can be installed to the workstation to access the selected language. You can then display data in the installed language by setting the active language preference code in the user profiles.

If you are installing more than one alternate language, the last language installed will be the language available with the standard package. If you want the ability to access more than one language within the same package, you need to build a custom package that includes all languages.

- Language text is stored in the Central Objects, System, Control Table, and Data Dictionary data sources. The data is loaded to the Deployment Server during the installation and copied to the Central Objects and the other data sources through the language installation process.

In OneWorld, J.D. Edwards provides 17 languages. You can run single-byte languages on single-byte or double-byte operating systems, but you must run double-byte languages only on double-byte operating systems.

The installation of AS/400 secondary languages has no effect on the multi-language capability of OneWorld, because AS/400 secondary languages are not used. Therefore, you can use any or all of the OneWorld multiple languages with only Primary English installed on your AS/400 system.

Language preference codes

The language preference codes are the standard language codes used throughout the software. The language preference code field located in the OneWorld user profile specifies which language is presented on forms or reports. All users are assigned a language preference code within the user profile.

A blank language preference code specifies the base language of English. Alternate language records always have a value in the language preference code.

The alternate language text tables have records that contain language preference codes. These tables can contain several records for each item, such as a user defined code value. There may be multiple records for a code value, with each code representing a different language.

If the language preference code does not have a corresponding translated language record, the system defaults to the base English record.

17.2.1.2 Database character set and code page considerations

Data within a database has a set of parameters to define it. Each character within the database is identified by a specific language preference code value. A collection of characters within a defined database is called a character set or code page. A character set or code page setting is a scheme for encoding character data. Every character is defined by a unique hexadecimal value. These values may change between databases and languages. Every language is

represented by at least one character set. Some character sets may contain multiple languages.

For example, the Western European character set contains all characters common to the Western European languages (Aa to Zz), and all special characters unique to the specific languages, such as Á, á, Ü, and ü. Asian character sets are all specific to one language.

Within OneWorld, J.D. Edwards uses code page conversions to control the consistent or desired display of data. A code page conversion adjusts the hexadecimal values of different characters so that the appearance of text on the desktop is the same with different code pages.

Code page settings

When installing the database, you must set up the code page for the language before loading your language specifics for OneWorld.

When using multinational code pages, J.D. Edwards has the following recommended settings for the IBM AS/400 system and workstations:

- **Database**

- For single-byte languages, J.D. Edwards recommends that the IBM U.S. English Code Page CCSID value should be set at 37 or the IBM Multinational Code Page CCSID value should be set at 500.
- For Japanese, J.D. Edwards recommends that the IBM Japanese National Code Page CCSID value should be set at 939 or 5026.

- **Workstations**

Code page settings for individual languages are specified by Microsoft in Windows 95, Windows 98, and Windows NT environments. The Microsoft Access database works with all code pages.

17.2.1.3 OneWorld national language support

Setting up NLS correctly in OneWorld environment, includes setting up code pages and jde.ini files for the Enterprise Server and workstations.

For example, you can enter French data on a Microsoft Windows-based workstation, which will be converted to EBCDIC on the IBM AS/400 system. The data automatically converts through IBM Client Access Express. The text is stored in a specific character set that uniquely describes the data as French. The same database can store alternate language text along with French, relying on NLS standards to manage the text storage and retrieval. J.D. Edwards uses NLS on all supported platforms, so you can interact with any computer system (hardware and software) within your own environment.

By using NLS standards, J.D. Edwards maintains consistent data within all databases and hardware platforms.

Code page and jde.ini values

In order for the workstation to reflect the language installed on the Deployment Server, you must perform tasks for both the Enterprise Server and workstations in addition to the procedures to verify and modify the jde.ini settings.

Single-byte and double-byte considerations

Many single-byte languages support either national code pages or multinational code pages. The double-byte languages support specific individual national code pages by language.

Single-byte character sets use a collection of phonetic characters that require one byte to create a single character. Conversely, the double-byte character sets use ideographic characters and require two bytes to create a single character.

Single-byte languages can generally be run on single- or double-byte systems. Double-byte languages, such as Japanese, Chinese, and Korean, must run on machines configured to support a double-byte system.

OneWorld can perform a Query By Example (QBE) on any character. However, when using a double-byte language, this process may not necessarily use an understandable collating sequence, although you can still use QBE for any double-byte column.

For workstation-to-AS/400 session software, J.D. Edwards requires a 32-bit ODBC driver if you are using a double-byte language. J.D. Edwards recommends IBM Client Access Express (32-bit ODBC driver), which supports both single- and double-byte languages.

Font considerations

J.D. Edwards includes standard fonts, by language, in a separate file. Some languages (such as double-byte) require certain fonts to display and print correctly. OneWorld software stores the font settings in files according to language. Individual OneWorld users can select fonts by language for forms, grids, and reports.

User display preferences

User display preferences are individually defined sets of OneWorld characteristics that are stored in the user profile. OneWorld uses these preferences to determine how menus and forms are presented to individual users and where language is to be used in OneWorld for that user. After user display preferences are set up for a given user, the values remain the same on any workstation that the user accesses.

17.2.1.4 CCSID issues

When running PORTTEST after installing OneWorld on the AS/400 Enterprise Server (refer to “Performing PORTTEST” on page 233), failures can occur if there are incorrect code page settings. To ensure the correct installation and running of OneWorld, ensure that the following settings are correct:

- Ensure that, if the value of the system value QCCSID on your system is not 37, the CCSID value of user profile QSECOFR is set at 37 during the LODRUN part of the install (refer to 9.1.1.8, “Installing libraries and Data Dictionary on the AS/400 system” on page 178).

By default, the CCSID value is set to *SYSVAL. If it is not changed at this time, it would impact the CCSID settings of the tables created at this time. Although this is not serious for the data tables such as Central Objects tables, it has a bigger impact on the TAM files created in the integrated file system (IFS) and all of the service programs (*SRVPGM) and modules (*MODULE).

- Ensure that OneWorld is installed on the AS/400 Enterprise Server using the JDE user profile with a CCSID of 37.
- Ensure that the CCSID specified in the jde.ini file on the AS/400 system (member JDE in the INI file in library B7332SYS) is also correctly set.
- Ensure that the CCSID for user profile ONEWORLD is set to 37.
- Ensure that the jde.ini file on the AS/400 system has a LocalCodeSet value of US_EBCDIC, not WE_EBCDIC.

17.2.2 Language installation

The option to include alternate languages is incorporated into the installation plan procedures. Languages can be installed using the Installation Planner. The Language CD is required to implement this functionality.

You can incorporate one or more languages either into your initial installation or into your update or upgrade plan. You can set up a language-only plan after the base installation takes place for an installation. You can also build a language-specific package.

From the languages you install, you can choose which languages will be installed for each selected environment using a custom plan. The control tables, Data Dictionary, and central objects will be loaded by the Control Table Workbench and the Specification Table Merge Workbench.

17.2.2.1 Language installation disk space requirements

Prior to performing your language installation, you must ensure that the affected machines have adequate disk space. Note that the space requirements for single-byte and double-byte languages are the same. Table 39 shows the requirements to load one alternate language.

Table 39. Alternate language disk space requirements

OneWorld machine	Estimated disk space requirements (per language)
Deployment Server ¹	250 MB
Client Workstation	100 MB
AS/400 Database	300 MB
Central objects for Pristine path code (Central Objects - PRISTB733)	
Central objects for Production path code (Central Objects - PRODB733)	
Central objects for CRP path code (Central Objects - CRPB733)	
1. The disk space requirements shown do not include the disk space requirements for the actual building (on the Deployment Server) of the package required. Those disk requirements (approximately 1.5 GB) are additional.	

17.2.3 Installing an alternate language during OneWorld installation

This section covers the process used to install an alternate language during the initial OneWorld installation. References are made to Chapter 9, "OneWorld

installation and maintenance” on page 151, for the corresponding steps in the installation process.

17.2.3.1 Installing an alternate language on the Deployment Server

Following the installation of platform independent objects (refer to 9.1.1.6, “Installing platform-independent objects” on page 174), you can install an alternate language on the Deployment Server.

Installing an alternate language on the Deployment Server performs the following tasks:

- The language.mdb file for the language you are installing is copied to the \Planner\Data directory.
- The spec.cab file replaces the spec.cab file in the selected path code’s \package directory.
- The jdeb7.mdb replaces the jdeb7.mdb in the selected path code’s \package\Data directory.
- OneWorld creates a new registry key (“LANGUAGE”) in the HKEY_LOCAL_MACHINE\SOFTWARE\JDEEDWARDS\OneWorld\Install.ini\B733 directory and String value for the language installed and its version. The registry key and String value are used in conjunction with the table F984031 - Language and Environment Plan Detail (stored in the jdeplan.mdb) during the creation of the update, upgrade, installation, and language-only installation plans.

To install an alternate language on the Deployment Server, complete the following steps:

1. Sign on the Deployment Server as JDE.
2. Insert the language CD into the AS/400 CD-ROM drive.
3. If the setup does not start automatically, use Windows NT Explorer to find the CD drive and click it. Find and double-click the **setup.exe** file for the language. A display appears like the example in Figure 254.

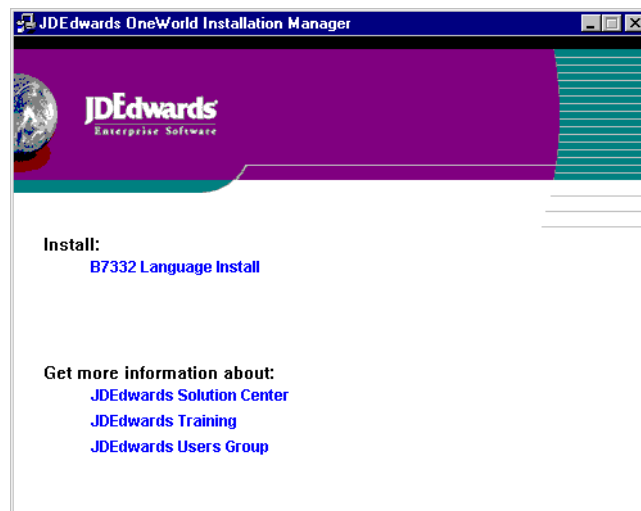


Figure 254. OneWorld language installation

4. Click **B7332 Language Install**.

- Click **Next** on the OneWorld installation program Welcome. The OneWorld Deployment Server Setup Type form appears as shown in Figure 255.

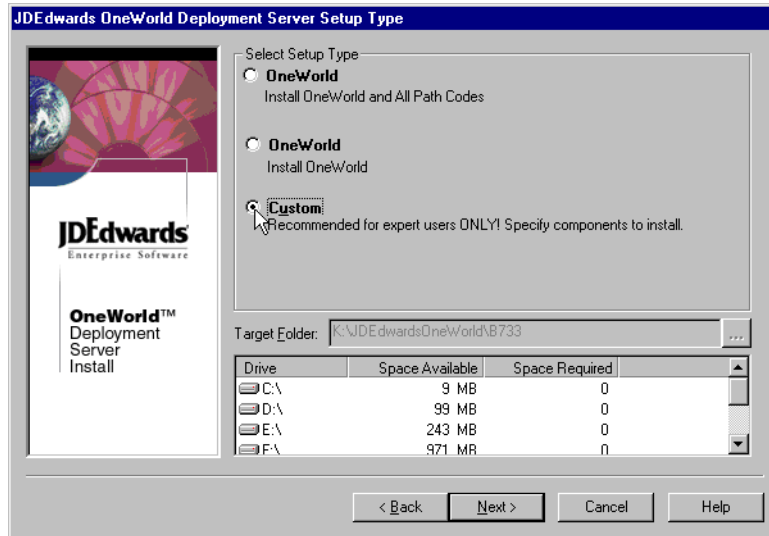


Figure 255. OneWorld Deployment Server Setup Type

- Click **Custom** for Select Setup Type. The target folder correctly shows the directory where OneWorld is installed.
- Click **Next**. You see a form that appears as shown in Figure 256.

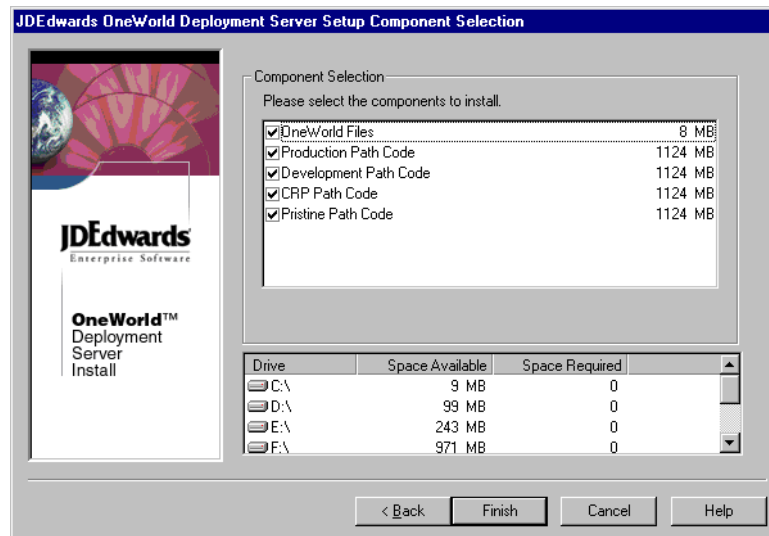


Figure 256. OneWorld Deployment Server Setup Component Selection

- Ensure that you select **OneWorld Files** and all path codes where you want to install the alternate language. Click **Finish**.

The JDEdwards OneWorld Deployment Server Setup Progress form appears as the files are loaded.

- Click **OK** when you see the message stating that the JDEdwards OneWorld Deployment Server Setup succeeded.

10. Cancel out of the OneWorld Installation Manager, because the installation of the language CD is now finished.

You can now continue with 9.1.1.7, “Sharing the directory \JDEdwardsOneWorld B73.3” on page 177.

17.2.3.2 Entering information into Installation Planner

After starting Installation Planner, you enter basic plan information, such as plan name and description, installation type, and included languages, if any (refer to “Entering information into Installation Planner” on page 190).

After you complete step 5 on page 190, in which you type B7332 in the To Release field, tab out of this field to Include Languages. Click **Yes**. Your display should appear similar to the example in Figure 257.

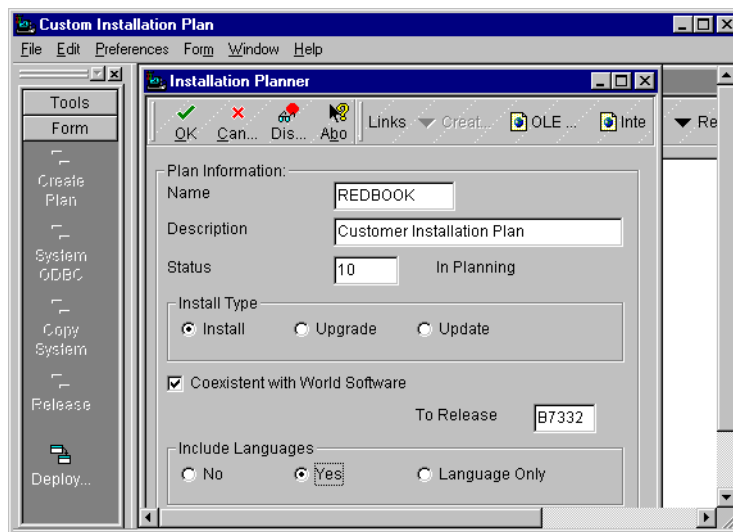


Figure 257. OneWorld Installation planner

Continue with step 6 on page 190.

17.2.3.3 Setting up environments

To specify default or custom selections for your installation environments, data load options, advanced environment parameters, and environment languages, you complete the tasks “Setting up environments and environment data sources” on page 204. In step 4 on page 204, ensure that you deselect **Default Languages**, as shown in Figure 73 on page 205.

To select language environments

Following step 7 on page 207, the Language Selection form appears. Complete the following steps:

1. Double-click your alternate language (ensure that the pink check mark appears). Your display should look similar to the example in Figure 258 on page 488.

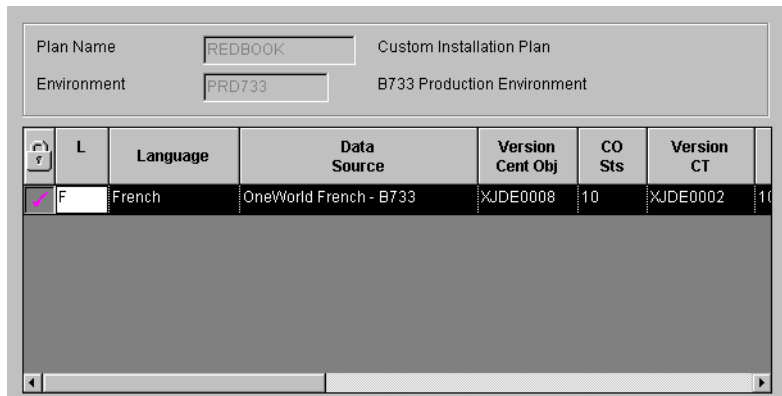


Figure 258. OneWorld language selection

2. Click **OK**.

Continue with step 8 on page 207.

17.2.3.4 Running Installation Workbench

When you run the Installation Workbench (refer to 9.1.3.4, “Installation Workbench” on page 213), the following workbenches are used for multiple languages:

- **Control Table Workbench:** Runs the batch application for the planned control table merges to update language files.
- **Specification Table Merge Workbench:** Runs the batch applications for the planned merges to update language files. This workbench runs only for users who are adding an alternate language to their installations.

Configuring your control tables

The distributed language tables consist of control tables, system tables, and Data Dictionary tables. The UDC tables contain only the desired language text, while the menu text and Data Dictionary files add the desired language on top of the already-installed English records.

Follow the process in “Configuring your machines” on page 220. You configure your Control Tables using the following steps:

1. On Control Table Workbench, OneWorld displays all control table merges in the detail area. On the Control Table Workbench form, the Plan Name field contains the name of your language plan. Your display should look similar to the example in Figure 259.

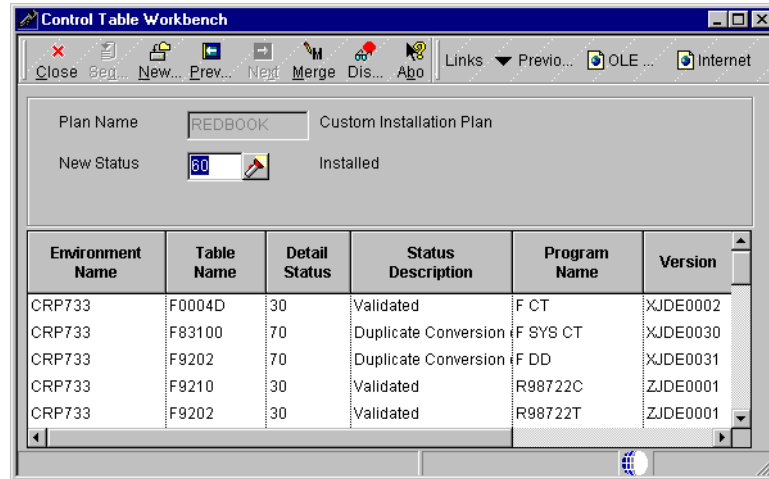


Figure 259. OneWorld Control Table Workbench

2. If you want to perform one merge at a time, select the appropriate row, and then click **Merge** on the Form menu. Otherwise, click **Merge All** on the Form menu.
3. After all merges complete, click **Next**.

Merging your specification tables

Users who are loading an alternate language with the installation must load language records into the central objects tables to enable that language.

The central objects tables contain the alternate language records needed for displaying text in the selected language. Language-enabled tables include the Processing Option Text table (F98306), the Report Design Aid Text Information table (F98760), and the Forms Design Aid Text Information table (F98750). Depending upon the environment choices during the English installation, you might have several sets of central objects, for example, one set for each environment loaded during the English-language installation.

To merge your specification tables, complete the following tasks:

1. On the Specification Table Merge Workbench form (similar to the example in Figure 260 on page 490), the Plan Name field contains the name of your language custom installation plan.
2. On Specification Table Merge Workbench form, click **Merge All** from the Form menu.

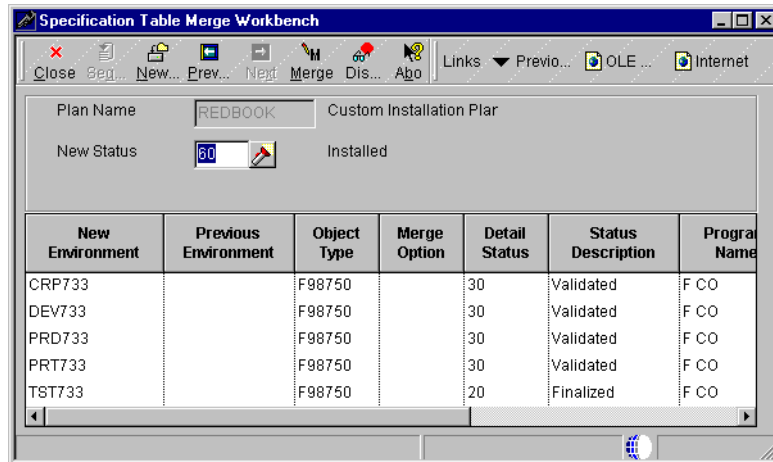


Figure 260. OneWorld Specification Table Merge

3. After the merge finishes, verify the output of the report that is produced.
4. After you have verified the output from the specification table merge, click **Next**.

Continue with “Configuring your packages” on page 221.

17.2.3.5 Installing OneWorld on the Enterprise Server

After making the other required changes, you must modify the Enterprise Server jde.ini file on the Deployment Server appropriately for your language (refer to 9.1.4.6, “Modifying the Enterprise Server jde.ini file” on page 226).

Verify the LocalCodeSet statement. In the [INSTALL] section, verify that your LocalCodeSet parameter is set to the appropriate value, based on your requirements. Also, verify that your code page is set up properly. For appropriate values and code pages, see Table 41 on page 505 and Table 42 on page 507. These values are set automatically, with the first language installed. For multiple languages, the first language, alphabetically, is the default and may need to be modified.

17.2.3.6 Copying language specifications to the Enterprise Server

You must load the language specification files to the Enterprise Server to create reports and to print in your desired language. The replicated local specification files that currently reside on your Enterprise Server contain the English language. The replicated local specification files that contain your languages are rdaspec, rdatext, potext, dddict, and ddtex.

The final step in preparing your Enterprise Server for the use of an alternate language is to copy the language specification files using the package management process. Note that you must build and deploy a separate package for each path code for which you need more than the base language.

To include language specifications in your package, you need to specify the language. The package build process then uses the language preference code specified as a parameter when building the package using the relational database tables to build the Form Design Aid and Report Design Aid specifications. A language package can be a full, update, or partial package.

When you build the package, the current Data Dictionary is included in the package.

Complete the following steps to build a full package (for more information on building and deploying OneWorld packages, refer to Chapter 13, “OneWorld package management” on page 327, and the *Package Management Guide*):

1. To define the package assembly, complete the following tasks:
 - a. Access the Work with Packages form by double-clicking **Package Assembly** from the Package and Deployment Tools menu (GH9083).
 - b. Click **Add** to launch the Package Assembly Director.
 - c. At the Welcome window, click **Next**.
 - d. At the Package Information form, type the package name, description, and associated pathcode. Click **Next**.
 - e. At the Package Type Selection form, select **Full**, and click **Next**.
 - f. Click **Next** on the Foundation Component form.
 - g. Click **Next** on the Help Component form.
 - h. Click **Next** on the Database Component form.
 - i. Click **Next** on the Default Object Component form.
 - j. At the Language Component form, double-click **Domestic** and each of the other languages you are installing. Ensure that the selected languages have a check mark. Ensure that you deselect **Selected Languages Only**. Click **Next**.
 - k. At the Package Components Revisions form, click **End**.
 - l. At the Work With Packages form again, expand **Packages**, so you can see the language package just created.
 - m. Activate the package assembly you just created by clicking the package name of the language package that was just created and clicking **Activate**.
2. To build the client and server packages and to deploy the server package, complete the following tasks:
 - a. Access the Work With Package Build form by double-clicking **Package Build (P9621)** from the Package and Deployment Tools menu (GH9083).
 - b. Click **Add** to launch the Package Build Definition Director.
 - c. Click **Next** at the Welcome window.
 - d. At the Package Selection form, click the package you already defined. Click **Next**.
 - e. At the Package Build Location form, click **Client**, and click **Server**. Click **Next**.
 - f. On the Server Selection form, select your Enterprise Server by double-clicking its name. Make sure a check mark is displayed by its name. Your display should look similar to the example in Figure 261 on page 492, except that your AS/400 Enterprise Server will be checked.


	Server Name	Data Source Type	Database Name
	AS400		
	Dec Alpha		
<input checked="" type="checkbox"/>	DENAS9	I	DENAS9 - B733 Server Map
	DENAS9 - Logic	I	DENAS9 - B733 Server Map
	HP9000		
	LOCAL	A	LOCAL
	OS390		
	RS6000		
	SNI		
	SUN		

Figure 261. OneWorld Package Build server selection

- g. At the Build Specification Options form, click **Build Specification Options**. Deselect **Individual Specification Tables**, because you are building a full package with all specifications.
- h. Click **Next**.
- i. On the Business Function Option form, click **Build Functions Options**. Type 20 (optimize) for Build Mode. Click **Next**.
- j. On the Compression Options form, select **Compression Options**. This reduces transfer time when the package is deployed (note, however, that this also increases package build time). This is necessary for the client package to ensure that the current compressed package is replaced before deploying. Click **Next**.
- k. At the Package Build Revisions form, verify your settings and make any necessary changes. Click **End** to finish the package build definition.
- l. At the Work With Package Build Definition form, click **Find** to see the build definition for your package.
- m. Activate your package build definition by clicking your language package and then clicking **Activate/Inactivate**.
- n. Expand your language package. Expand **Client** and expand your AS/400 server. Expand **Properties** within both Client and your AS/400 server. Verify the package build definition has a build status of Definition Complete.
- o. When you are ready to initiate the build process, click the language package name to highlight it, and click **Submit Build** from the Row menu.
- p. After the build completes, verify that all components were built successfully. If not, you need to determine the cause and resolve it before continuing. To help you determine the cause of the build failing, review the following files:
 - **JDE.LOG file:** Usually found on your C: disk in the root directory, but may be found in another location based on your jde.ini file.
 - **BUILDERROR.TXT file:** Found in your package directory, for example, the k:\JDEwardsOneWorld\B733\pathcode\PACKAGE\packagename\ directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the

environment for which the multiple languages are being installed, and *packagename* is the name of your language package.

- **BUILDLOG.TXT file:** Found in your package work directory, for example, the k:\JDEDwardsOneWorld\B733\pathcode\PACKAGE\packagename\WORK directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the environment for which the multiple languages are being installed, and *packagename* is the name of your language package.
 - **BUILDREPORT.TXT file:** Found in your package work directory, for example, the k:\JDEDwardsOneWorld\B733\pathcode\PACKAGE\packagename\WORK directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the environment for which the multiple languages are being installed, and *packagename* is the name of your language package.
- q. To deploy the server package, access the Work with Package Deployment form by double-clicking **Package Deployment (P9631)** from the Package and Deployment Tools menu (GH9083).
 - r. Click **Packages**, and click **Find** to see the language server package you want to deploy. Click your package to highlight it.
 - s. Click **Deploy** from the Row menu.
 - t. Click **Next** on the Welcome display.
 - u. Click your package name to highlight it, and click **Next**.
 - v. Click **Enterprise Server**, and click **Next**.
 - w. Double-click your AS/400 server name (ensure that a check mark appears). Click **Next**.

17.2.3.7 Installing the workstations

OneWorld software is first installed to the Deployment Server, and then it is deployed from the Deployment Server to workstations. There are several methods to install OneWorld to workstations.

In order for the workstation to reflect the language installed on the Deployment Server, you must perform tasks for both the Enterprise Server and workstations, in addition to the procedures to verify and modify the *jde.ini* settings.

Since the language package that you built in 17.2.3.6, “Copying language specifications to the Enterprise Server” on page 490, is a full package, it can be deployed either to an already installed OneWorld client workstation to add the new language or to a new workstation that does not already have OneWorld installed.

For instructions on installing workstations, refer to 9.1.5, “Configuring and installing client workstations” on page 236. Ensure that you select your language package name in step 13 on page 238.

User profile and language considerations

All users are assigned a language preference code within the user profile. The language preference code field specifies which language is presented on the

applicable form or report. You can set up your users and groups when you determine how you want groups to be organized, and what preferences you want each group to have. For more information on creating or modifying OneWorld user profiles, refer to 15.2.1, “OneWorld user profiles and group profiles” on page 421.

For languages, the `jde.ini` files are updated automatically. For multiple language usage, both the server and workstation `jde.ini` files need to be modified for the preferred language being installed.

17.2.4 Creating a language-only installation plan

Use the procedures in this section when installing one or more additional languages when OneWorld has been previously installed.

17.2.4.1 Installing an alternate language on the Deployment Server

Tasks performed by the Deployment Server during the installation of the alternate language are described in 17.2.3.1, “Installing an alternate language on the Deployment Server” on page 485.

We recommend that, before you begin creating your language-only Installation, you make a copy of your `spec.cab` file in the selected path code’s `\package` directory, the `jdeb7.mdb` file in the selected path code’s `\package\Data` directory, and the `JDEPLAN.MDB` file in the `\Planner\Data` directory. This protects your Deployment Server installation in the event that problems occur in other steps of the installation process. If the original files are corrupted later, you have backup copies of these files.

1. If you do not already have one, create a temporary directory for these backups.
2. Using Windows NT Explorer, copy the **JDEB7.MDB** file from the `n:\JDEdwardsOneWorld\B733\pathcode\package\Data` directory to the temporary directory, where *n* is the drive location of the OneWorld software on the Deployment Server, *pathcode* is the path code corresponding to the environment in which the alternate language is being installed, and *package* is the name of your language-only installation plan.
3. Using Windows NT Explorer, copy the **JDEPLAN.MDB** file from the `n:\JDEdwardsOneWorld\B733\Planner\Data` directory to the temporary directory, where *n* is the drive location of the OneWorld software on the Deployment Server.

To install an alternate language on the Deployment Server, complete the following steps for each alternate language that you want to install:

1. Sign on the Deployment Server as `JDE`.
2. Insert the language CD into the AS/400 CD-ROM drive.
3. If setup does not start automatically, use Windows NT Explorer to find the CD drive and click on it. Find and double-click the **setup.exe** file for the language.
You see a display like the example in Figure 254 on page 485.
4. Click **B7332 Language Install**.

5. Click **Next** on the OneWorld installation program Welcome. The OneWorld Deployment Server Setup Type form appears as shown in Figure 255 on page 486.
6. Click **Custom** for Select Setup Type. The target folder correctly shows the directory where OneWorld is installed.
7. Click **Next**. You see the form shown in Figure 256 on page 486.
8. Ensure that you select **OneWorld Files** and the all path codes where you want to install the alternate language. Click **Finish**.
The J.D. Edwards OneWorld Deployment Server Setup Progress form appears as files are loaded.
9. Click **OK** when you see the message stating that the J.D. Edwards OneWorld Deployment Server Setup succeeded.
10. Cancel out of the OneWorld Installation Manager, because the installation of the language CD is now finished.

17.2.4.2 Creating a custom installation plan for languages

To enter information into the installation planner, complete the following steps:

1. Sign on to the Deployment Server in the planner environment (JDEPLAN) of J.D. Edwards OneWorld as user `JDE` with the password `JDE`.
2. From the System Installation Tools menu (GH961), double-click **Custom Installation Plan**.
3. On the Work with Installation Plans form, click **Add**.
4. On the Installation Planner form, type basic plan information by completing the following fields:
 - **Name:** The name of your language-only installation plan.
 - **Description:** The description of your language-only installation plan. Provide as much detail as possible.
 - **Status:** Type 10 as the status of your language-only installation plan.
 - **Install Type:** Click **Install** as the installation type.
 - **Coexistence:** Select **Coexistence** if you currently have OneWorld installed in coexistence with WorldSoftware. Otherwise, deselect **Coexistence**.
 - **To Release:** In the To Release field, type `B7332`. Tab out of this field.
 - **Include Languages:** Select **Language Only** as the option of your plan.
5. Click **OK**.

Installation Planner provides you the opportunity of either defining a new location or selecting one already defined. Complete the following tasks:

1. When prompted to associate your language-only plan with a location, click **Select** to select an existing location.
2. Select your base location (the location with blank Location Code and Parent Location fields).
3. Click **OK**.
4. At the Location Revisions form, click **OK**.

Installation Planner displays the Environment Selection form.

1. At the Environment Selection form, deselect **Default Environments**, and deselect **Default Languages**.
2. Click **OK**. Installation Planner displays the Select Environments form.
3. Click to highlight the environment where you want to install a language, and click **Select**.
4. Installation Planner displays the Language Selection form. Double-click each language to install. A check mark appears in the first column to acknowledge the selection. Your display should look similar to the example in Figure 262.

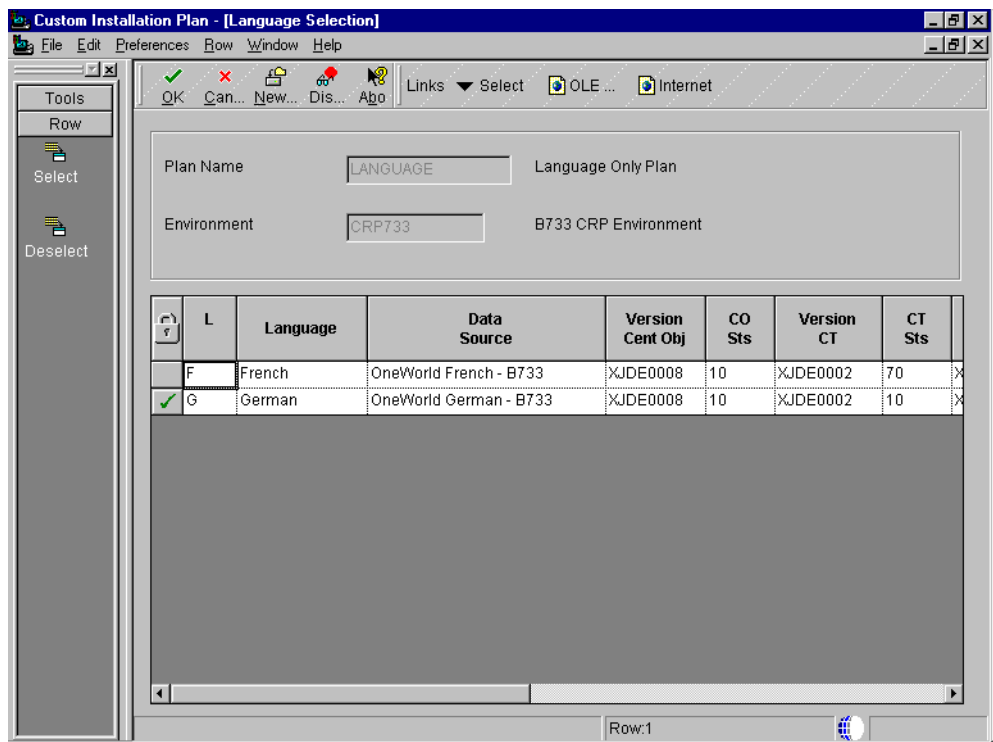


Figure 262. OneWorld Language Selection

5. Click **OK**.
Installation Planner displays the Select Environments form again.
6. If you have another environment where you want to install another language, return to step 3 on page 495.
Note: If you decide to include multiple environments, you need to delete some duplicate items from your plan after it is finished (but before it is validated).
7. After you finish the definition of environments, click **Close** at the Select Environments form.
8. Installation Planner asks if you want to add another location. Click **No**.
9. When Installation Planner displays the Congratulations screen, click **OK**.
Installation Planner automatically starts the Work with Batch Versions program.

Note

If you included multiple environments, cancel the Plan Validation Report program. In Work with Installation Plans, highlight and expand your language-only plan. Under environments, delete the following duplicate entries under Control Table Merges for all but one environment: F83100 (x SYS CT) and F9202 (x DD), where x is the language being installed.

If an environment listed shares Central Objects with another environment included in the plan, also delete the duplicate F98750 (x CO) under Specification Merges.

After deleting these items, run **Validation** from the Row menu.

Figure 263 shows a custom language-only plan.

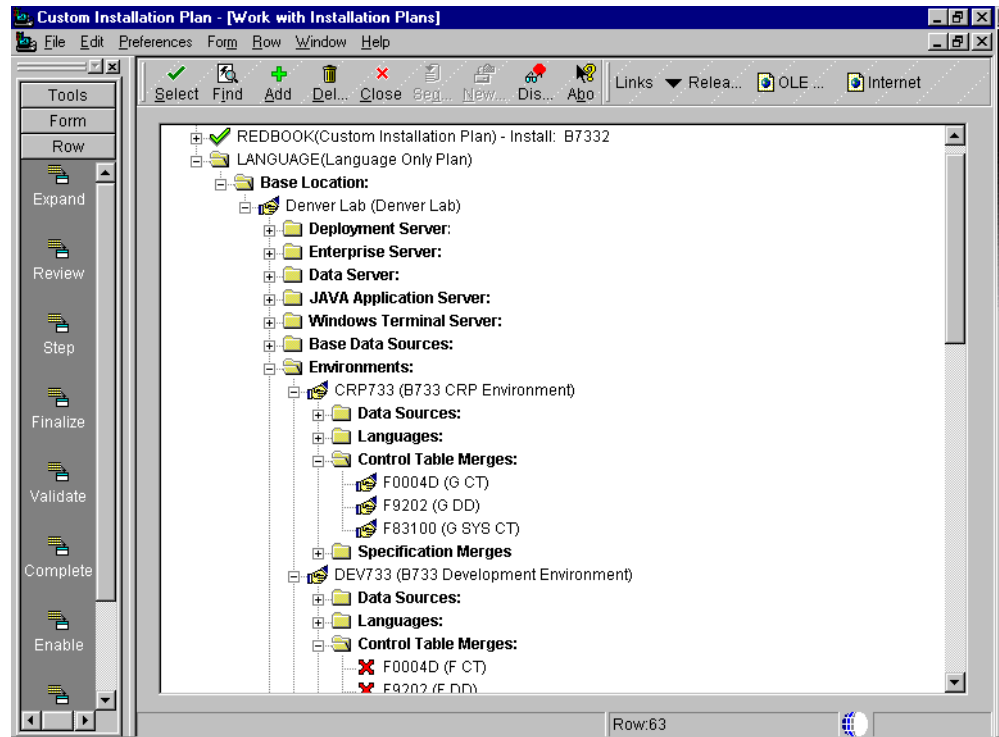


Figure 263. OneWorld Language Only Custom Installation Plan

10. Double-click Version **XJDE0001** to initiate the batch application.
11. Select **Data Selection** on the Version Prompting form.
12. Click **Submit**.
13. Click “**TESTPLAN**” on the Data Selection form.
14. Double-click **<Literal>**.
15. Type your custom language-only installation plan name.
16. Click **OK**.
17. The Data Selection form re-appears with your plan name shown in the Right Operand. Click **OK**.

The Processing Options Extra Info tab appears. Chapter 7 of the *OneWorld Installation Reference Guide (All Systems)* describes, in detail, the available processing options and suggests which settings may be suitable.

18. Type **Y** for Additional data source information flag.
19. Type **D** for OCM Information Flag.
20. Type **N** for OCM Differences Flag.
21. Type **Y** for Path Code Information Flag.
22. Type **30** for Update the Plan Status.
23. Click **OK**.
24. The Report Output Destination form appears. Select **On Screen**.
25. Click **OK**.
26. Review the resulting R9840B Validation Planner Report to verify that your installation plan is correct. Validation Results of Record Validated and Duplicate Entry are normal. Other Validation results should be investigated before continuing with the Installation Workbench.
27. On the Work With Installation Plans form, expand your plan and each part of the plan to review.

17.2.4.3 Running Installation Workbench for a language-only plan

After you plan the OneWorld language installation, you run the language-only plan using another OneWorld application, Installation Workbench.

For a language-only installation, the following workbenches are used:

- Control Table Workbench copies the language text from the Microsoft Access database that contains the language text into the control table database and central object database.
- Specification Table Merge Workbench runs the batch applications for the planned merges to update language files.

To start Installation Workbench, follow these steps:

1. Sign on to the Deployment Server in the planner environment (JDEPLAN) of J.D. Edwards OneWorld as user **JDE** with the password **JDE**.
2. From the System Installation Tools menu (GH961), double-click **Installation Workbench**.
3. On the Work with Installation Plan form, double-click your language-only plan. Installation Workbench displays the Control Table Workbench form. The Plan Name field contains the name of your language-only plan.

On the Control Table Workbench form, OneWorld displays all control table merges in the detail area. Your display should look similar to the example in Figure 259 on page 489.
4. If you want to do one merge at a time, select the appropriate row, and then click **Merge** on the Form menu. Otherwise, click **Merge All** on the Form menu.
5. After all merges complete, verify the output reports produced.
6. Click **Next**.

7. Installation Workbench displays the Specification Table Merge Workbench form.
8. On the Specification Table Merge Workbench form (similar to Figure 260 on page 490), the Plan Name field contains the name of your language plan.
9. On the Specification Table Merge Workbench form, click **Merge All** from the Form menu.
10. After the merge finishes, verify the output of the report that is produced.
11. After you verify the output from the specification table merge, click **Next**.
12. Installation Workbench displays the Installation Completed Successfully! screen. Click **Finish**.

17.2.4.4 Building an alternate language package

A package represents a copy of the central objects at a point in time. It contains replicated objects which OneWorld reads at run-time. If custom modifications or text overrides have been made at the time a software update is to be performed, or if you want to deploy development changes to a local workstation including a specific language, you need to build a package and specify which language or languages to include in that package. This involves the following considerations:

- Package build considerations
- Package deployment considerations

Package build considerations

Before you begin to build a language package, verify that your language installation is complete. To build the language package, you must first define the package. This procedure can take several hours depending on the size of the package and the number of languages used. This task is completed on the Deployment Server, on which you sign on as user JDE in DEPB733.

To include language specifications in your package, you need to specify the language, which determines the language to include. The package build process then uses the language preference code specified as a parameter when building the package using the relational database tables to build the form design aid text and report design aid text specifications. A language package can be a full, update, or partial package. When you build the package, the current Data Dictionary is included in the package.

To include your UDC and menu language changes in a package, they must be replicated to the Access database that is included in the package.

Package deployment considerations

You cannot deploy a translated package to a workstation if the appropriate character set is not installed on that workstation. For example, if you created a package containing Japanese text, the workstation must be loaded with the Japanese version of Windows 95, Windows 98, or Windows NT to view the Japanese data correctly.

To transfer translated objects to a server, complete the server package installation procedures. You need to define each object that you modified for the languages. Therefore, you must track the objects you changed so that you can include them in a package.

For more information on building and deploying packages, refer to Chapter 13, “OneWorld package management” on page 327, as well as the *Package Management Guide*.

Updating specification files on the Enterprise Server

After you run the Installation Workbench for your language-only plan, you must create a server package to update the Data Dictionary items and specification files on the Enterprise server. This package will be of type full and will include all specification tables.

The process of updating the specification files on the Enterprise server consists of three steps: assembly of package, definition of server package build, actual build, and deployment of server package.

Complete the following steps to build a full package (for more information on building and deploying OneWorld packages, refer to Chapter 13, “OneWorld package management” on page 327, as well as the *Package Management Guide*):

1. To define the package assembly, complete the following steps:
 - a. Access the Work with Packages form by double-clicking **Package Assembly** from the Package and Deployment Tools menu (GH9083).
 - b. Click **Add** to launch the Package Assembly Director.
 - c. On the Welcome display, click **Next**.
 - d. On the Package Information form, type the package name, description, and associated path code. Click **Next**.
 - e. On the Package Type Selection form, select **Full**, and click **Next**.
 - f. Click **Next** on the Foundation Component form.
 - g. Click **Next** on the Help Component form.
 - h. Click **Next** on the Database Component form.
 - i. Click **Next** on the Default Object Component form.
 - j. On the Language Component form, double-click **Domestic** and each of the other languages that you previously installed, as well as those that you are now installing. Ensure that the selected languages have a check mark. Ensure that you deselect **Selected Languages Only**. Click **Next**.
 - k. On the Package Components Revisions form, click **End**.
 - l. On the Work With Packages form again, expand **Packages**, so you can see the language package just created.
 - m. Activate the package assembly you created by clicking the package name of the language package just created and clicking **Activate**.
2. To build the client and server packages and deploy the server package, complete the following steps:
 - a. Access the Work With Package Build form by double-clicking **Package Build (P9621)** from the Package and Deployment Tools menu (GH9083).
 - b. Click **Add** to launch the Package Build Definition Director.
 - c. Click **Next** at the Welcome screen.
 - d. On the Package Selection form, click the package you already defined. Click **Next**.

- e. On the Package Build Location form, click **Client**, and click **Server**. Click **Next**.
- f. On the Server Selection form, select your Enterprise server by double-clicking its name. Make sure a check mark is displayed by its name. Your display should look similar to the example in Figure 261 on page 492, except that your AS/400 Enterprise Server will be selected.
- g. On the Build Specification Options form, click **Build Specification Options**. Deselect **Individual Specification Tables**, because you are building a full package with all specifications.
- h. Click **Next**.
- i. On the Business Function Option form, click **Build Functions Options**. Type 20 (optimize) for Build Mode. Click **Next**.
- j. On the Compression Options form, select **Compression Options**. This reduces transfer time when the package is deployed (note, however, that this will also increase package build time). This is necessary for the client package to ensure that the current compressed package is replaced before deploying. Click **Next**.
- k. On the Package Build Revisions form, verify your settings and make any necessary changes. Click **End** to finish the package build definition.
- l. On the Work With Package Build Definition form, click **Find** to see the build definition for your package.
- m. Activate your package build definition by clicking your language package and then clicking **Activate/Inactivate**.
- n. Expand your language package. Expand **Client** and expand your AS/400 server. Expand **Properties** within both Client and your AS/400 server. Verify the package build definition has a build status of Definition Complete.
- o. When you are ready to initiate the build process, click the language package name to highlight it, and click **Submit Build** from the Row menu.
- p. After the build completes, verify that all components were built successfully. If not, you need to determine the cause and resolve it before continuing. To help you determine the cause of the build failing, review the following files:
 - **JDE.LOG file**: Usually found on your C: disk in the root directory, but may be found in another location based on your jde.ini file.
 - **BUILDERROR.TXT file**: Found in your package directory, for example, the k:\JDEwardsOneWorld\B733\pathcode\PACKAGE\packagename\ directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the environment for which the multiple languages are being installed, and *packagename* is the name of your language package.
 - **BUILDLOG.TXT file**: Found in your package work directory, for example, the k:\JDEwardsOneWorld\B733\pathcode\PACKAGE\packagename\WORK directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the environment for which the multiple languages are

being installed, and *packagename* is the name of your language package.

- **BUILDREPORT.TXT file:** Is found in your package work directory, for example, the k:\JDEdwardsOneWorld\B733\pathcode\PACKAGE\packagename\WORK directory, where *k* is the drive where OneWorld is installed on your Deployment Server, *pathcode* is the path code corresponding to the environment for which the multiple languages are being installed, and *packagename* is the name of your language package.
- q. To deploy the server package, access the Work with Package Deployment form by double-clicking **Package Deployment (P9631)** from the Package and Deployment Tools menu (GH9083).
- r. Click **Packages**, and click **Find** to see the language server package you want to deploy. Click your package to highlight it.
- s. Click **Deploy** from the Row menu.
- t. Click **Next** on the Welcome display.
- u. Click your package name to highlight it, and click **Next**.
- v. Click **Enterprise Server**, and click **Next**.
- w. Double-click your AS/400 server name (ensure that a check mark appears). Click **Next**.

Updating the UDC and menu tables

During Installation Workbench, the control table merges updated your master user defined code and menus tables in the Control Tables data source (for example, Control Tables - CRP) with the information from the change tables provided by J.D. Edwards in the language CDs. Your master menu tables were updated with any menus you added in the previous release. You must now copy the updated UDC and menu tables into the JDEB7.MDB Access database used to deploy workstations.

For users installing an alternate language, this procedure merges the language control tables text with the control tables. The following procedure should be completed after Installation Workbench, before deploying the alternate language to client workstations. It updates the JDEB7.mdb Access database that gets deployed to client workstations with the language records added to the Control Tables.

To update the UDC and menu tables, follow these steps:

1. On the Deployment server, sign on to the OneWorld Planner (JDEPLAN) environment as user `JDE` with password `JDE`.
2. Type `GH9011` in Fast Path, and press Enter.
3. From the System Administration Tools menu (GH9011), double-click **Batch Versions**.
4. Type `R98403` in Batch Application, and click **Find**.
5. Run the appropriate versions from the following list. Initially, you should run the version in Proof mode to make sure it is setup properly. Repeat this step for each version that you need to run.

- Production Environment (non-coexistence):
XJDE0010 Production Control Table Copy
 - Production Environment (coexistence):
 - XJDE0013 AS/400 Common - Production - UDC
 - XJDE0014 AS/400 Common - Production - Menu Tables
 - Test Environment (non-coexistence):
XJDE0011 Test Control Table Copy
 - Test Environment (coexistence):
 - XJDE0015 AS/400 Common - Test - UDC
 - XJDE0016 AS/400 Common - Test - Menu Tables
 - CRP Environment (non-coexistence):
XJDE0012 CRP Control Table Copy
 - CRP Environment (coexistence):
 - XJDE0017 AS/400 Common - CRP - UDC
 - XJDE0018 AS/400 Common - CRP - Menu Tables
- a. Double-click the version you want to run.
 - b. On the Version Prompting form, click **Submit**.
 - c. On the Processing Options form, click **Update** tab. In field 6, type 1 to run in Proof mode (this should be done initially to ensure that it is what you want, as it does not make any changes to your database). In field 6, type 2 to Create the Environment Database. Leave all other fields at their default value. Click **OK**.
 - d. On the Report Output Destination form, click **On Screen**, and click **OK**.

Deploying alternate languages to the client workstations

In order for the workstations to reflect the language installed on the Deployment Server, you must perform tasks for both the Enterprise Server and workstations, in addition to the procedures to verify and modify the `jde.ini` settings. Since the language package that you built is a full package, it can be deployed either to an already installed OneWorld client workstation to add the new language or to a new workstation that does not already have OneWorld installed.

For instructions on installing workstations, refer to 9.1.5, “Configuring and installing client workstations” on page 236. Ensure that you select your language package name in step 13 on page 238.

User profile and language considerations

All users are assigned a language preference code within the user profile. The language preference code field specifies which language is presented on the applicable form or report. You can set up your users and groups when you determine how you want groups to be organized, and what preferences you want each group to have. For more information on creating or modifying OneWorld user profiles, refer to 15.2.1, “OneWorld user profiles and group profiles” on page 421.

For languages, the `jde.ini` files are updated automatically. For multiple language usage, both the server and workstation `jde.ini` files need to be modified for the preferred language being installed.

17.2.5 Post-installation language verification

Following the installation of an alternate language in OneWorld, you should verify that the language records have been merged into the necessary Control Tables on the AS/400 system. On an AS/400 command line, type `STRSQL` and press Enter.

For each OneWorld path code (for example, `PRODCTL`, `CRPCTL`, `DEVCTL`, `TESTCTL`), type each of the following SQL statements and press Enter after each statement, where `xxxx` is `PROD`, `CRP`, `TEST`, and `y` is the language (for example, `F` for French, `G` for German):

```
Select count(*) from xxxxCTL.F0004D where DTLNGP = 'y'
Select count(*) from xxxxCTL.F0005D where DRLNGP = 'y'
Select count(*) from xxxxCTL.F0083 where MTLNGP = 'y'
Select count(*) from xxxxCTL.F98800D where PKLNGP = 'y'
Select count(*) from xxxxCTL.F98810D where AKLNGP = 'y'
```

For each OneWorld path code (for example, `PRODB733`, `CRPB733`, `PRISTB733`, `DEVB733`), type each of the following corresponding SQL statements and press Enter after each statement, where `xxxx` is `PROD`, `CRP`, `PRIST` or `DEV`, and `y` is the language (for example, `F` for French, `G` for German):

```
Select count(*) from xxxxB733.F98750 where FTLNGP = 'y'
Select count(*) from xxxxB733.F98760 where RTLNGP = 'y'
Select count(*) from xxxxB733.F98306 where PTLNGP = 'y'
```

For each OneWorld release (for example, `B733`), type each of the following corresponding SQL statements and press Enter after each statement, where `y` is the language (for example, `F` for French, `G` for German):

```
Select count(*) from SYSB733.F83100 where DTLNGP = 'y'
Select count(*) from SYSB733.F83110 where CHLNGP = 'y'
Select count(*) from SYSB733.F91100D where FDLNGP = 'y'
Select count(*) from DDB733.F9202 where FRLNGP = 'y'
Select count(*) from DDB733.F9203 where FRLNGP = 'y'
Select count(*) from DDB733.F00165 where GDTXKY like '%|y|%'
```

Table 40 shows the results of running these SQL statements. If your observed record count is approximately the same as the record count listed, you can feel confident that you have successfully performed the Control Table and Specification merges on the AS/400 system.

Table 40. Multi-language control tables record count

Table	Language	Approximate number of records	Language	Approximate number of records
xxxxCTL/F0004D	French (F)	2541	German (G)	3207
xxxxCTL/F0005D	French (F)	20590	German (G)	23998
xxxxCTL/F0083	French (F)	9507	German (G)	6901
xxxxCTL/F98800D	French (F)	855	German (G)	828
xxxxCTL/F98810D	French (F)	3536	German (G)	3536
xxxxCTL/F98750	French (F)	76329	German (G)	75045
xxxxCTL/F98760	French (F)	27890	German (G)	28478
xxxxCTL/F98306	French (F)	12189	German (G)	12031

Table	Language	Approximate number of records	Language	Approximate number of records
SYSB733/F83100	French (F)	4	German (G)	4
SYSB733/F83110	French (F)	1	German (G)	1
SYSB733/F91100D	French (F)	46	German (G)	46
DDB733/F9202	French (F)	34687	German (G)	34687
DDB733/F9203	French (F)	58280	German (G)	58280
DDB733/F00165	French (F)	37858	German (G)	37046

17.3 Tips and hints

This section includes tips and hints for implementing SBCS and DBCS languages in OneWorld.

17.3.1 Tips for implementing OneWorld SBCS languages

Table 41 contains the LocalCodeSet values for the Enterprise Server jde.ini files and the code page single-byte settings for OS/400. OneWorld supports the following single-byte (Western European) languages: U.S. English, Canadian English, United Kingdom English, Australian, Portuguese, German, French, Spanish, and Italian. The single-byte settings are shown in Table 41.

Table 41. Single-byte LocalCodeSet and code page settings

Language	LocalCodeSet	Windows 95, 98, and Windows NT	OS/400
U.S. English, Canada English, Australian, and Portuguese	WE_ISO88591	1252	N/A
German	DE_EBCDIC	N/A	273
Spanish	ES_EBCDIC	N/A	284
French	FR_EBCDIC	N/A	297
Italian	IT_EBCDIC	N/A	280
U.K. English	UK_EBCDIC	N/A	285
U.S. English, U.K. English, Canada English, French, Canadian French, German, Spanish, and Italian	US_EBCDIC	N/A	37

Language	LocalCodeSet	Windows 95, 98, and Windows NT	OS/400
U.S. English, U.K. English, Canada English, Australian, Canadian French, Portuguese, French, German, Spanish, and Italian	WE_EBCDIC	N/A	500

17.3.1.1 Tested SBCS implementation

We installed French Canadian, French Standard, and German Standard keyboard languages on a United States English Windows NT Client Workstation with OneWorld installed. The AS/400 system has a primary language of English. The system value QCCSID has a value of 65535. The user profile QSECOFR uses the default *SYSVAL for the CCSID parameter. The user profile JDE uses 37 (U.S. English) for the CCSID parameter.

With these settings, all of the tables within the OneWorld Business Data data sources are created with CCSID 37. CCSID 37 appears to provide the capability for supporting all single byte characters in the J.D. Edwards tier 1 languages.

The OneWorld English, French, and German languages have been installed. We installed French with English during the OneWorld installation process and added German using a Language-Only Installation Plan. Based on the language preference in the OneWorld user profiles, we can successfully view displays and produce reports in English, French, and German.

Characters keyed into OneWorld forms using any of the English, French, French Canadian, and German keyboards on the Windows NT Client Workstation display correctly when keyed. These metalanguage characters also correctly display when they are retrieved. As well, these characters are correctly stored in the AS/400 OneWorld data libraries using CCSID 37.

When double-clicking the OneWorld icon on the workstation, the language in which the OneWorld signon display appears is based on the Initial_Language_Code statement in the [Interactive Runtime Section] of the jde.ini file on the client workstation. After signing on, the language used for displays and reports is based on the language specified in the OneWorld user profile used to sign on. This signon automatically updates the Initial_Language_Code statement in the jde.ini file with the language of this OneWorld user.

Changing the CCSID value for individual AS/400 user profiles corresponding to the OneWorld users has no apparent effect on the way that characters are stored in the OneWorld AS/400 data libraries. Shutting down OneWorld, changing the CCSID of user profile JDE (for example, to 297), and restarting OneWorld also has no apparent effect on the way that characters are stored in the OneWorld AS/400 data libraries. The only parameter that appears to affect the coding of characters in the OneWorld AS/400 data libraries is the CCSID of the tables (or individual columns in the tables). Unless the CCSID of these tables has been altered (this is *not* recommended), the CCSID for these tables should be correctly specified based on the correct value of user profile JDE at the time of installation of OneWorld.

Based on our testing, the J.D. Edwards recommendation of using CCSID 37 for user profile JDE when installing OneWorld appears to provide the required capability required for supporting multiple tier 1 J.D. Edwards languages.

It is also possible that other CCSIDs for national or multinational languages (for example, multinational CCSID 500 and French national CCSID 297) may work successfully when specified in user profile JDE during OneWorld installation. For other CCSIDs to work correctly, it is necessary that they provide the capability for all characters that will be keyed on any other keyboard language required. It is also necessary that any other CCSID used provide the capability for any special characters (for example #, @, \$, and others) required by J.D. Edwards. However, this was not tested. Therefore, if you decide to use a CCSID other than 500, do so at your own risk while knowing that J.D. Edwards recommends CCSID 37 be used.

17.3.2 Tips for implementing OneWorld DBCS languages

At this time, J.D. Edwards supports the installation of OneWorld with only one double-byte language for any environment. If you have a requirement for using more than one double-byte language but do not need to share the data between these multiple double-byte environments, you should consider configuring your AS/400 for Logical Partitioning (LPAR). For more information on Logical Partitioning, see 2.1.3, “Object-based architecture” on page 11. This provides you with the ability to configure independent logical AS/400 partitions running OneWorld, with a single double-byte language being used in each logical partition.

Table 42 contains the LocalCodeSet values for the Enterprise Server jde.ini files and the code page double-byte settings for OS/400.

OneWorld supports the following double-byte languages: Japanese, Traditional Chinese (Taiwan and Hong Kong), Simplified Chinese (China and Singapore), and Korean. The double-byte settings are shown in Table 42. J.D. Edwards supports the Double-Byte Character Set (DBCS) on Windows 95, 98, and NT.

Table 42. Double-byte LocalCodeSet and code page settings

Language	LocalCodeSet	Windows 95, 98, and Windows NT	OS/400
English Uppercase DBCS and Japanese (Katakana DBCS)	JA_SJIS	932	N/A
English Uppercase DBCS and Japanese (Katakana DBCS)	JA_EBCDIC	N/A	939
English Uppercase and Lowercase DBCS and Traditional Chinese (Taiwan, Hong Kong)	TC_BIG5	950	N/A
English Uppercase and Lowercase DBCS and Traditional Chinese (Taiwan, Hong Kong)	TC_EBCDIC	N/A	937
English Uppercase and Lowercase DBCS and Simplified Chinese (China, Singapore)	SC_GB	936	N/A

Language	LocalCodeSet	Windows 95, 98, and Windows NT	OS/400
English Uppercase and Lowercase DBCS and Simplified Chinese (China, Singapore)	SC_EBCDIC	N/A	935
English Uppercase and Lowercase DBCS and Korean	KO_KSC	949	N/A
English Uppercase and Lowercase DBCS and Korean	KO_EBCDIC	N/A	933

17.3.2.1 Creating the Processing Option Text Table

To add or retrieve data from the Processing Option Text table (F98306), double-byte customers must recreate this table in their specified language ID format. Double-byte languages include Japanese, traditional Chinese, simplified Chinese, and Korean.

This procedure must be performed after loading the Central Objects (refer to 9.1.2, "Installing the OneWorld databases" on page 183):

1. Sign on to the AS/400 system. Type `QSECOFR` for User and its associated password. Then, press Enter.
2. Confirm that the CCSID in the JDE user profile is set to the correct character set ID, for example, CCSID=939 (Japanese Language):
 - a. Type `DSPUSRPRF JDE` and press Enter. Scroll down until you see the value for the CCSID parameter.
 - b. If the CCSID value for JDE is *SYSVAL, verify the value of system value QCCSID. Type the command `DSPSYSVAL QCCSID` and press Enter.
 - c. If the CCSID value of JDE is not correct or if it is *SYSVAL and the QCCSID system value is not correct, correct the value of the CCSID in user profile JDE. Type `CHGUSRPRF JDE CCSID(xxx)` and press Enter, where xxx is the required CCSID value.
3. Sign onto the Deployment Server's OneWorld Planner environment (JDEPLAN) using the user ID `JDE` and the password `JDE`.
4. Type `OL` for Fast Path, and press Enter to start the Object Librarian application.
5. Type `F98306` in Object Name, and click **Find**.
6. Select the record for **F98306**.
7. Complete the following steps for each Central Objects data source:
 - a. From the Form Exit menu, select the **Copy Table** option.
 - b. Select **OneWorld Local** for the source data source.
 - c. Select your Central Objects data source (for example, **Central Objects - CRPB733**) for the destination.
 - d. Create the ODBC connection for the Central Objects Library (refer to "OneWorld and ODBC Data Sources" on page 191). Confirm that the Central Objects data source is correctly set, as this process drops the F98306 table in the Central Objects Library and creates a new format of the table with the character set ID.
 - e. Run this **Copy Table** for the Central Objects library you have installed.

- f. Use the following AS/400 command to verify the new table's CCSID:

```
DSPFFD YYYYY/F98306
```

Here, YYYYY is the Central Objects Library name, such as COCRPB733.

17.3.2.2 B732 OneWorld double-byte installation

Consider the following points when performing a OneWorld double-byte installation:

- IBM user profiles JDE, ONEWORLD, and QSECOFR must have the correct CCSID specified before the installation workbench is run so that all data files have the correct field CCSID, for example, 937 for Traditional Chinese. Note that you must use other CCSID values for other languages, for example, 935 for Simplified Chinese. The AS/400 QCCSID system value can remain at 65535.

To make this change, type `CHGUSRPRF USRPRF(user) CCSID(value)` on an AS/400 command line and press Enter. Here, *user* is JDE, ONEWORLD, or QSECOFR, and *value* is the correct double-byte language CCSID value.

- All AS/400 user profiles, corresponding to the OneWorld users who need to deal with double-byte characters (for example, to enter a double-byte Alpha Name in the Address Book Revision program), must have the correct CCSID setup also.

To make this change, type `CHGUSRPRF USRPRF(user) CCSID(value)` on an AS/400 command line and press Enter. Here, *user* is the OneWorld user needing DBCS capability, and *value* is the correct double byte language CCSID value.

- Modify Client Access ODBC in the Windows NT registry. In Client Access for Windows 95/NT and Client Access Express, a registry entry needs to be created with a string value of `DBCSNoTruncErr=1`.

If there is no `DBCSNoTruncErr` or `DBCSNoTruncErr=0` (default), the 32-bit driver returns a truncation error, since IBM believes that some applications want to be informed about this error. However, this causes SQL errors with OneWorld. If this entry is not correct, this will affect language installation and the double-byte characters that are not copied from the Microsoft Access database to the AS/400 files.

To make this change so that the 32-bit Client Access ODBC driver behaves the same as the 16-bit ODBC driver, complete the following steps:

1. Click **Start->Run**.
2. Type `regedit.exe` and click **OK**.
3. Expand **HKEY_LOCAL_MACHINE**. Expand **SOFTWARE**. Expand **ODBC**. Expand **ODBC.INI**.
4. Repeat the following steps for each Client Access ODBC Data Source (for example, Business Data - PROD):
 - a. Click the appropriate Client Access ODBC Data Source (for example, Business Data - PROD).
 - b. The status bar at the bottom shows `MyComputer\HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\odbcdatasource`, where *odbcdatasource* is the name of your selected ODBC Data Source.

- c. Right-click the right-hand pane of the window (not on an entry, but elsewhere in the window), or click **Edit**.
- d. Click **New**. Click **String Value**.
- e. Type `DBCSNoTruncErr` for the Name, and press Enter.
- f. Double-click the string value **DBCSNoTruncErr** you just created.
The *Edit String* window pops up.
- g. Type `1` in the Value data box.
- h. Click **OK**.
- i. Exit the Regedit program.

A similar correction needs to be done on the Deployment Server as well, since this will carry the correct ODBC setting to new client workstations when installed:

1. Edit the `ODBCDataSource.INF` file found in the `\JDEdwardsOneWorld\B733\Client` folder. Add the last line as shown in the following example.

Note: You must add this line to each Client Access ODBC Data Source, for example:

```
[Business Data - CRP]
Driver=E:\Program Files\IBM\Client Access\Shared\cwboDBC.dll
.....
Decimal=0
DBCSNoTruncErr=1
```

2. Ensure that the `jde.ini` files are setup correctly. Failure to have this setting will result in a Communication Error when a UBE is submitted:
 - **AS/400 Server:** Change member JDE in the INI file in the B7332SYS library (this is an example for Traditional Chinese, so make the appropriate changes corresponding to the DBCS language that you are using):

```
[INSTALL]
Double_Byte=0
LocalCodeSet=TC_EBCDIC
```

- **Client PC:** Change `jde.ini` file in the `\winnt` folder:

```
[INSTALL]
Double_Byte=0
LocalCodeSet=TC_BIG5
```

17.4 OneWorld localization

Localization is the process of enhancing the base J.D. Edwards software to meet country-specific statutory requirements and common business practices within the strategic industries and markets that J.D. Edwards currently targets.

Globalization is the process of enhancing the base J.D. Edwards software to meet global business requirements within the strategic industries and markets that J.D. Edwards currently targets. J.D. Edwards provides a consistent quality product, installed via the standard J.D. Edwards process, with discrete program numbers, and is documented and translated in a manner coordinated with J.D. Edwards release schedules.

Part 3. Faster, smoother, easier...

This part covers some advanced topics that will be interesting for those of you who want to optimize your OneWorld AS/400 system to run smoother, faster, easier, better... This part includes these chapters:

- Chapter 18, “The Integrated Netfinity Server and OneWorld” on page 513, introduces you to the Integrated Netfinity Server. This chapter describes how the Integrated Netfinity Server works and uses AS/400 resources, as well as how it functions in a OneWorld environment.
- Chapter 19, “Data replication and data migration” on page 529, discusses data replication and data migration, along with some of the issues and considerations involved when choosing to replicate or migrate your data.
- Chapter 20, “Server-to-server communication” on page 545, describes the choices available for connecting two or more AS/400 OneWorld servers, for example a OneWorld Application Server to a OneWorld Database Server. This chapter is particularly beneficial to those of you who want to implement OneWorld servers on more than one system.
- Chapter 21, “Performance management” on page 561, introduces you to a performance management methodology for optimally running the OneWorld application on the AS/400 system.
- Chapter 22, “Problem identification and solution” on page 711, will help you identify and manage problems that you may encounter when running OneWorld on the AS/400 system. It focuses on identifying the source of the problems and possible solutions.

Chapter 18. The Integrated Netfinity Server and OneWorld

This chapter gives a brief introduction to the Integrated Netfinity Server and describes how it works and uses the AS/400 resources. It includes sections on using the Integrated Netfinity Server as a OneWorld Deployment Server as well as a Windows NT Terminal server. This chapter also discusses startup and shutdown procedures.

For more detailed information on the Integrated Netfinity Server, refer to *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164, or visit the Web page at: <http://www.as400.ibm.com/NT>

18.1 Introduction to the Integrated Netfinity Server

All AS/400 RISC models are capable of accommodating the Integrated Netfinity Server. There are several types of Integrated Netfinity Servers, which differ in capacity and processing power.

AS/400 Models 270 and 8XX can accommodate the Integrated Netfinity Server with an Intel 700 MHz Pentium III microprocessor with up to 4 GB of on-board memory, 2 Universal Serial Bus (USB) ports, and a video chip.

AS/400 Models 170 and 7XX can accommodate the Integrated Netfinity Server with an Intel 333 MHz Pentium II microprocessor with up to 1024 MB of on-board memory

The above AS/400 models can have a maximum of three on-board LAN adapters supporting Token-Ring or Ethernet (10 MB or 100 MB). Up to 16 Integrated Netfinity Servers can be installed in the AS/400 system depending on the AS/400 model.

Figure 264 shows a photograph of the 700 MHz Integrated Netfinity Server.

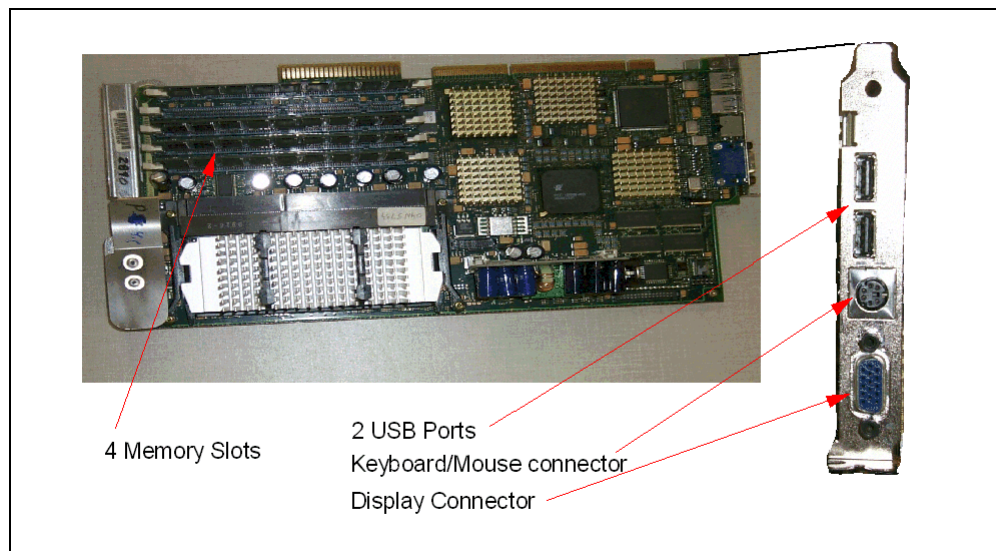


Figure 264. 700 MHz Integrated Netfinity Server

A display, keyboard, and mouse are connected directly to the Integrated Netfinity Server to provide a Windows NT 4.0 or a Windows 2000 Server console. OS/400 V4R5 is required to run Windows 2000 Server on the Integrated Netfinity Server. Windows 2000 Advanced Server and Windows 2000 Data Center Server are currently not supported on Integrated Netfinity Server. Parallel and serial ports are also available on the Integrated Netfinity Server for the direct attachment of devices, such as a modem or printer.

Windows NT or a Windows 2000 Server running on the Integrated Netfinity Server have access to AS/400 disks, tape units, and CD-ROM drive. AS/400 disk resources are allocated to the Windows NT/2000 server (up to a maximum of 128 GB per server). They automatically take advantage of the AS/400 system RAID-5 and mirroring capabilities (if configured).

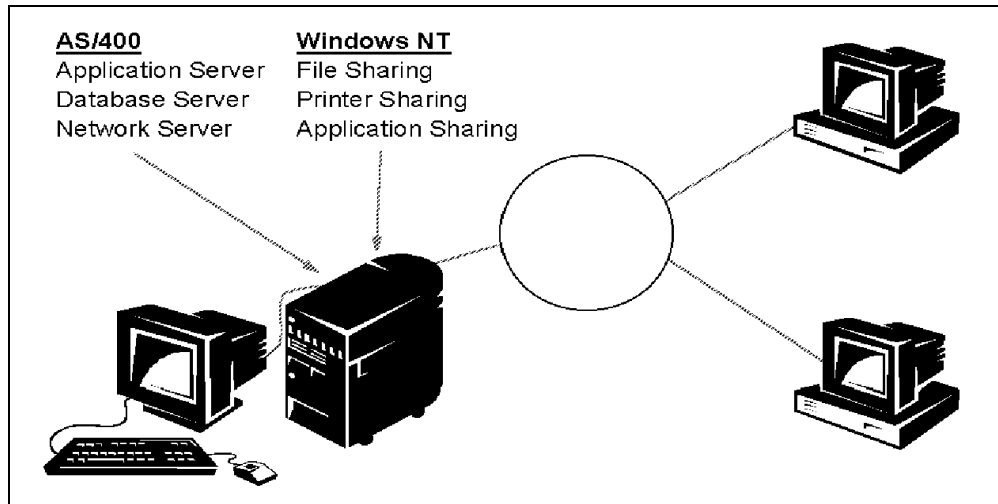


Figure 265. AS/400 integration with the Windows NT/2000 server

OS/400 V4R5 provides support for file-level backup and restore. Support for file-level backup in previous OS/400 releases is provided through Windows interfaces, giving Windows NT/2000 server backup applications direct access to AS/400 tape units. In addition, OS/400 backup commands can be used for disaster recovery, and ADSM/400 can provide an alternative for file-level backup and restore.

The Integrated Netfinity Server's LAN adapters can be used for access to both Windows NT/2000 Server and the AS/400 system (for example, using AS/400 Client Access). An internal "LAN", which is really a bus connection between the Windows NT Server and the AS/400 system, provides a secure connection protected from external network disruptions. This connection is used for a variety of functions including installation, user administration, and DB2 UDB for AS/400 access.

Using the integrated administration functions of AS/400 Integration with Windows NT/2000 server, customers can reduce the overhead of maintaining two sets of user profiles. An AS/400 operator can create or delete both an AS/400 user and a Windows NT/2000 user in a single operation. Users can change their password once on the AS/400 system and have the change automatically propagated to the Windows NT/2000 server.

AS/400 Client Support for Windows NT allows the use of Microsoft's System Message Block (SMB) protocol to enable AS/400 system applications written to the integrated file system APIs to access data on the Windows NT server. In V4R2, AS/400 NetServer support enabled AS/400 files to be viewed in the Network Neighborhood. This support used the System Message Block functions.

The QNTC file system is accessible from AS/400 commands, Client Access network drives, or Operations Navigator. This support works with Microsoft Windows NT/2000 installed on the Integrated Netfinity Server or on a standalone PC server.

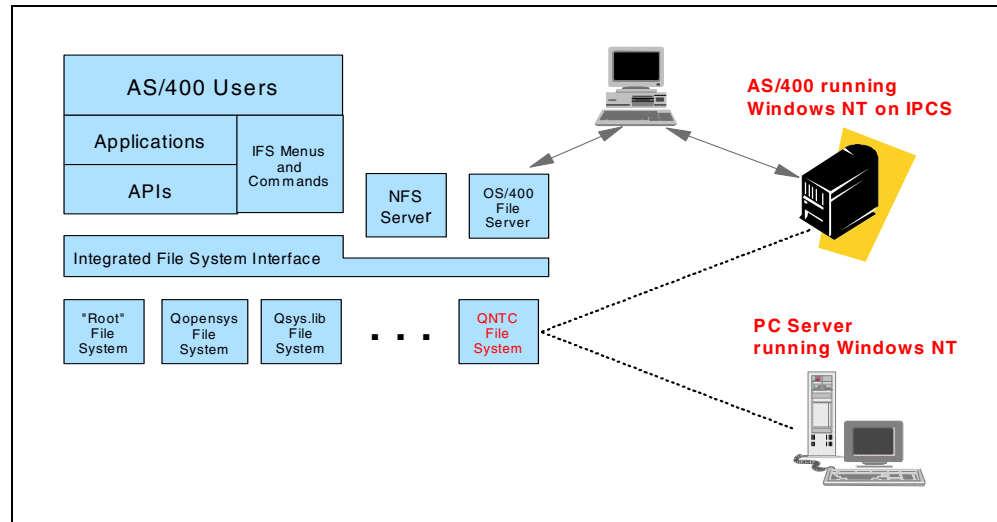


Figure 266. Example of the QNTC file system and integrated file system

OS/400 V4R5 provides an Operations Navigator plug-in for a graphical user interface that can be used for managing the Windows NT/2000 server on the Integrated Netfinity Server.

For more information on AS/400 Integration with Windows NT Server, refer to *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164. Also, visit the Web page at: <http://www.as400.ibm.com/NT>

18.1.1 OS/400 V4R5 enhancements

There are several enhancements for Integrated Netfinity Server available in OS/400 V4R5. They are summarized in the following list:

- New 700 Mhz Pentium III processor with 4 GB memory and two USB ports that boost performance
- File level backup allows saving and restoring the integrated file system (IFS) files with OS/400 commands SAV/RST.
- Server management GUI through the plug-in in the network component of the Operations Navigator. It allows to start/stop, display status, and display/change properties of Windows servers.
- Windows 2000 support through a no-charge licensed program 5769-WSV (option 29 of the OS/400). This licensed program is automatically shipped on the bonus pack with all OS/400 V4R5 orders.

- Storage space usage. In V4R5, the C: drive is used as a boot drive *and* it stores Windows, which is not the case with V4R4 (refer to “Drives and drive spaces” on page 520). This requires a significantly larger C: drive.
- Support for FAT32 file system (for Windows 2000, 98, 95) with a maximum storage space size of 32 GB.
- Increased maximum size of the NTFS storage space to 64000 GB.
- Total network storage space size increased to a maximum of 884.8 GB.
- Support for the Windows 2000 Plug & Play feature.

18.1.2 OS/400 V4R4 enhancements

A number of new features and enhancements to OS/400 Integration with Windows NT server are available with OS/400 V4R4. These were not available with the previous Integrated PC Server product. They include:

- **User auxiliary storage pool (ASP) support for network server storage:** Network server storage space support is enhanced to allow storage spaces to reside in either the system ASP or user ASP.
- **System drives up to 8000 MB:** The Install Windows NT Server (INSWNTSVR) command is enhanced. A Windows system drive size up to 8000 (MB) may be allocated to a Windows server. System drives that are allocated greater than 2047 MB are automatically converted to NTFS by the Windows installation.
- **Option to install from CD-ROM Source Image:** The Install Windows NT Server (INSWNTSVR) command is enhanced. You can now specify an arbitrary integrated file system (IFS) path for the Windows NT CD-ROM source image that is used to install a Windows NT server on an Integrated Netfinity server.
- **NWSD Configuration Files:** The Install Windows NT Server (INSWNTSVR) command is enhanced. You can now write a network server description (NWSD) configuration file that customizes the Windows NT installation process. You then specify the NWSD configuration file on the INSWNTSVR command to install a Windows NT server on an Integrated Netfinity server.

18.2 How the Integrated Netfinity Server works

This section describes how the Integrated Netfinity Server works with Windows *NT* and OS/400 V4R4.

The Integrated Netfinity Server consists of a PCI or SPD card, which plugs into the AS/400 system. On each card, there is an Intel processor, up to 4 GB of memory, up to three Network Interface cards, and connections for a keyboard, mouse, display, serial port, and parallel port. The Integrated Netfinity Server uses a bus level connection with the AS/400 system to use disk and other resources on the AS/400 system.

18.2.1 Using system resources

Even though the Integrated Netfinity Server contains its own processor and memory, there are tasks that are handled by the AS/400 processors and memory. These tasks involve access to the disks and other AS/400 resources.

The Integrated Netfinity Server uses the AS/400 system's fast disk subsystem. Because the disk input/output (I/O) processing is performed by the AS/400 I/O processors, and not by the Intel CPU, the Integrated Netfinity Server may perform the same as a higher-speed CPU external server.

AS/400 disk storage is allocated to Windows NT, creating a storage space object or area of AS/400 disk space from the system pool of disk resources. Because of the way that the AS/400 disk subsystem scatters data across available disks, the Windows NT storage space is automatically spread out over multiple disks and uses AS/400 RAID 5 protection (if implemented). With V4R4, Windows NT storage spaces can be located in user auxiliary storage pools. At times of peak usage, the Integrated Netfinity Server may impact disk subsystem performance, since it uses the same disk arms as the AS/400 system. To minimize this impact, it is important to ensure there are an adequate number of physical disks in the AS/400 auxiliary storage pool being used.

18.2.1.1 Using the LAN card

The Integrated Netfinity Server has up to three network interface cards, which can be Token Ring, Ethernet, or a combination of both. These cards can be used in several ways. They can be configured as a network card for AS/400 access, as a network card for Windows NT access, or as a dual purpose card to access both.

In most cases, we recommend that you use the card for Windows NT access only and configure a second card for AS/400 network access. When configured for dual access, the network interface is under the control of Windows NT. If the Integrated Netfinity Server is varied off, it cannot be used for AS/400 network access. Using the card for both Windows NT and AS/400 access may be suitable in smaller workgroup environments.

18.2.1.2 Communication with the AS/400 system

The Integrated Netfinity Server uses a fast bus connection with the AS/400 system, which acts as a virtual Token Ring between the AS/400 system and the Integrated Netfinity Server. This Token-Ring connection can be managed through TCP/IP configuration, and the line configuration, just as a physical network connection. Because the traffic does not travel on the physical, external network, this virtual Token Ring can be used as a secure, fast, dedicated connection to the AS/400 system from the Integrated Netfinity Server. This may be recommended in certain situations, such as environments with high network utilization, or when security is of highest concern. Although the Integrated Netfinity Server's traffic passes over the high-speed bus, it functions as a 16 Mb Token Ring. If the external LAN is a 100 Mb or 1 Gb Ethernet, in most cases, performance is best over the high speed LAN. For more detailed information and instructions on using the internal virtual Token Ring, refer to the redbook *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164.

18.3 Using OneWorld with the Integrated Netfinity Server

Because of the level of integration with the AS/400 system and the ability to use AS/400 system resources, such as disks and tape, the Integrated Netfinity Server is an excellent choice for many OneWorld implementations.

J.D. Edward's OneWorld requires the use of the Windows NT server for the Deployment Server. Windows NT Terminal Server Edition is also a popular

add-on for OneWorld installations. Using the Integrated Netfinity Server feature for the Deployment Server as well as other OneWorld servers can be beneficial in consolidating and managing your OneWorld networked environment.

For specific recommendations on use of the Integrated Netfinity Server as a Deployment server and Windows Terminal server, refer to 18.3.3, "Integrated Netfinity Server as a Deployment Server" on page 519, and 18.3.4, "Integrated Netfinity Server as Windows Terminal Server (WTS)" on page 521.

18.3.1 Benefits of Integrated Netfinity Server for OneWorld users

There are several distinct advantages that OneWorld users should consider when evaluating the potential for using the Integrated Netfinity Server:

- **Security:** All existing AS/400 users and groups are integrated into Windows NT security. This feature greatly enhances the administration tasks.

If a customer is using MS SQL with trusted security, the existing AS/400 users are incorporated into Windows NT and SQL. This greatly reduces the administration issues regarding multiple platforms when OneWorld Security Server is not used.

- **Server consolidation:** Up to 16 Windows NT servers are consolidated into one AS/400 system. OneWorld customers can have multiple Windows NT servers, which are centrally located and managed serve as Deployment Servers and Windows Terminal Servers.
- **Availability:** It is possible to have a hot spare server available if required. If an Integrated Netfinity Server card fails, it is not difficult to switch over to another card. All that is required is that the spare card has equivalent I/O and memory capabilities as the card that failed. You then simply vary off the Integrated Netfinity Server that failed, switch the LAN cable, change the hardware resource to the spare card, and vary it on.

The storage that is allocated to Windows NT servers on Integrated Netfinity Servers is spread across multiple disk storage arms. This improves the disk storage access performance for certain applications. If you use mirroring or RAID on an AS/400 system, the Windows NT server implicitly uses the same disk protection.

- **Simpler disk management:** With the AS/400 managing disks, it becomes a simple exercise in allocating more disk capacity to a server, even remotely. All that is required is that the disk space be available on the AS/400 system. To allocate it to the server, you simply create a new storage space, vary off the server, attach it to the server, vary it back on. Then, you sign on to Windows NT and format it. If you have an existing storage space that you want to move, you can unlink it from one server and link it to another server.
- **Improved system management:** Apart from managing the disks remotely, other management facilities are available. These include monitoring Windows NT Event Log messages in an AS/400 message queue, submitting Windows NT commands from the AS/400 system, using AS/400 backup and recovery procedures, integrating Windows NT and AS/400 user profiles, and remotely rebooting Windows NT.
- **More flexible testing:** Using a single physical Integrated Netfinity Server, you can define a test and production server. When you want to carry out testing,

you simply vary off the production server and vary on testing, which minimizes the risk to production.

- **Reliability:** Using AS/400 hardware increases system's reliability.
- **Reduced network workload (increased performance):** Because there is no competition from other network traffic, LAN communications between the Windows NT server and the AS/400 system can be faster (especially if the external LAN runs at 10 mbps) and more secure.
- **Cost of ownership:** With Integrated Netfinity Server, there is no need to purchase, install, and maintain a separate PC server, which means reduced costs and complexity of managing a PC network. Reduced costs also include sharing of system resources, such as CD-ROM, disks, and tape units.

18.3.2 Integrated Netfinity Server considerations

Generally, the Integrated Netfinity Server functions as any other Windows NT server running on an Intel processor. However, there are some factors to consider when implementing an Integrated Netfinity Server based solution with OneWorld:

- **AS/400 Integrated Netfinity Server contention:** During heavy interactive use and disk activity from both the AS/400 system and Windows NT, there is the potential for some resource contention. Consider using the new feature in the Integrated Netfinity Server that provides support for storage spaces, which can reside in either the system ASP or user ASPs.
- **Client Access performance:** Always ensure that the version of Client Access Express and service packs as defined in Informational APAR are proper for the OS/400 version that is currently being used.
- **Multiple processor support:** The Integrated Netfinity Server currently supports only one Intel processor per card. IBM is aware of this limitation and will provide a solution to remedy this problem.
- **Backup and recovery:** An AS/400 full system backup backs up the Integrated Netfinity Server drive spaces. Each drive space is backed up as an object and can only be restored as an entire object. This does not allow a single file from the drive space to be restored.

Since the Integrated Netfinity Server can access the AS/400 tape drive, file-level backup and restore can be done through Windows NT, either through the backup applet or through a third-party backup application, such as Backup Exec from Seagate.

- **Windows 2000:** When this redbook was published, the Integrated Netfinity Server support for Microsoft Windows 2000 was planned for third quarter in 2000. For up-to-date information on supported software, refer to the Web site:
<http://www.as400.ibm.com/windowsintegration>

18.3.3 Integrated Netfinity Server as a Deployment Server

The OneWorld Deployment Server is used as the location of the initial OneWorld install. The software is then installed to clients and other servers from the Deployment Server. The Deployment Server is also used to build new software packages to be sent to clients and servers whenever the OneWorld software is customized, changed, or updated.

The Integrated Netfinity Server can be used as a Deployment Server in smaller OneWorld environments. Because the Integrated Netfinity is currently limited to a

single processor and some of the deployment and package build tasks are processor intensive, the Integrated Netfinity may not be suitable (performance wise) for larger implementations or environments that require a large amount of customization or frequent changes.

To use the Integrated Netfinity Server as a OneWorld Deployment Server, a standard Microsoft Windows NT license is required.

18.3.3.1 Recommendations

This section contains recommendations for those who use the Integrated Netfinity Server as a Deployment Server on OS/400 V4R4.

Windows NT swap file

The default Windows NT swap file is probably too small for most installations. The current Microsoft recommendation for the Windows NT server is to change the swap file to be 120% of actual physical memory. For example, if the Integrated Netfinity in use has 512 MB of memory installed, the swap file should be changed to a minimum of 615 MB.

Physical memory

The OneWorld Deployment Server has a minimum requirement of 128 MB of physical memory. In many environments, this may not provide suitable performance. We recommend that the Deployment Server be configured with at least 512 MB of physical memory. For larger implementations, more memory is recommended.

Drives and drive spaces

The default installation of Integrated Netfinity with OS/400 V4R4 configures a C: drive of 10 MB, a D: drive of 200 MB, and an E: drive of 500 MB. The C: drive contains boot files and should not be altered. However, the default sizes of the D: and E: drives may be too small for most implementations.

The D: drive contains files for the installation CD. The standard size of 200 MB is suitable for Windows NT 4.0, but it may need to be changed for other versions of Windows.

The E: drive contains the actual Windows NT program files and is also where many applications prefer to install files. The initial swap file is also stored on the E: drive. Much of the default 500 MB is taken up by the Windows NT program files. If you plan to install other software applications to the same drive as Windows NT or use the E: drive to hold the swap file, the size of this drive should be increased at the time of the installation. The current recommendation is to create an initial drive size of at least 1 GB, install most other applications to a separate drive, and create a separate drive space for the swap file.

Due to the Integrated Netfinity Server's use of the AS/400 disk subsystem, there is no performance gain from using multiple drives or creating Windows NT volume sets. The AS/400 system already spreads data across multiple drives for the best performance. Volume sets can still be used on the Integrated Netfinity servers to group multiple storage spaces into one larger, logical Windows NT drive. Multiple storage spaces and multiple drives under Windows NT can also be used to help organize applications and files.

A good use of a separate storage space is to install a given application to a separate storage space. If an Integrated Netfinity server card fails, that storage

space can be linked to another Integrated Netfinity Server card, and the application is again available quickly.

To better organize drive space and allow easy increases in size, the swap file can also be placed in a separate storage space.

A typical OneWorld install requires roughly 16 GB of storage space on the Deployment Server. We generally double this amount to allow for upgrades and changes. Currently the Integrated Netfinity disk storage spaces are limited to 8000 MB. To create a logical drive with enough space, you can create multiple 8000 MB storage spaces and link them together as one logical drive using a volume set under the Windows NT disk administrator.

CD-ROM

Since adding drives and storage spaces is so easy with the Integrated Netfinity Server, it is a good idea to rename the CD-ROM drive to use a drive letter that will not be changed by adding other drives and not confuse your drive naming scheme. Through the Windows NT disk administrator, the CD-ROM drive can be renamed to a different drive letter, such as X: or Z:. This allows the use of drive letters earlier in the alphabet for logical disk drives.

18.3.4 Integrated Netfinity Server as Windows Terminal Server (WTS)

OneWorld B73.3 includes support for Windows NT 4.0 Terminal Server Edition (TSE). This new variation of Windows NT allows a number of client workstations to connect to the Enterprise Server through a single client connection. This enables network traffic to be kept to a minimum, since all ODBC data retrievals from the server are performed by the Windows Terminal Server. This relays the minimum required data, such as updated screens, to the client.

There are a few considerations when setting up Windows NT 4.0 TSE to support J.D. Edwards OneWorld. These are noted in the following sections. To use the Integrated Netfinity Server as a Windows Terminal Server, a Microsoft Windows terminal server license is required.

18.3.4.1 Environments

A typical OneWorld installation uses several standard environments, including Production, Conference Room Pilot, and Development. The standard environments are configured to process master business functions (application logic) on the client workstation. In a Windows Terminal Server environment, there are special environments, prefixed with a "W", which map master business functions to the Enterprise Server.

Attention

It is important that master business function logic process on the application server and UBEs are submitted to the Enterprise Server. Otherwise, if master business functions and UBEs are processed on the Terminal Server Spec file, corruption may occur.

For a more detailed discussion of OneWorld mappings, refer to Chapter 4, "Configurable Network Computing (CNC)" on page 47, or the *J.D. Edwards Installation Reference Guide*.

Note

At this time, the development cannot be done in a Windows Terminal Server OneWorld environment. When development is performed, the TAM files are opened and stored in local cache. If multiple users attempt to update the same files, Spec file corruption may occur.

18.3.4.2 Using Citrix with WTS

OneWorld supports the use of Citrix Metaframe. This third-party application runs on top of Windows Terminal Server and provides an improved communications protocol, called ICA. The ICA protocol compresses network traffic and offers decreased network utilization when compared with the native protocol used in Windows Terminal server. This is particularly beneficial in a WAN environment where bandwidth is much smaller than a LAN and network traffic becomes more critical.

Citrix Metaframe can be run on the Integrated Netfinity Server and should be recommended in a WAN environment or other situations where network traffic is critical.

18.3.4.3 Sizing

The number of Windows Terminal Server users that can be run on a given server depends greatly on the number and type of applications being run. For example, if a OneWorld environment is configured to run master business functions on the client, you can run far fewer users than if the environment is configured to run those functions on the Enterprise Server.

Also, if other third-party applications are being used, such as personal productivity suites or e-mail, performance will vary greatly.

The Integrated Netfinity, with a 333 Mhz processor and 1 GB of physical memory can support between 10 and 20 concurrent OneWorld users, depending on the OneWorld applications being used.

18.3.4.4 Recommendations

This section contains some recommendations for those who use Integrated Netfinity Server as a Deployment Server.

Installation of Client Access Express

The Windows Terminal Server uses Client Access Express to connect to the AS/400 system using ODBC. Generally Client Access Express can be installed by double-clicking the SETUP.EXE file. However, in a Windows Terminal Server environment, it is important to use the ADD/Remove Programs option in Control Panel, since this option allows the application to be installed for all or selected TSE users.

Windows swap file

The default Windows NT swap file is probably too small for most installations. The current Microsoft recommendation for Windows NT Terminal server is to change the swap file to be 240% of actual physical memory. For example, if the Integrated Netfinity Server in use has 1 GB of memory installed, the swap file should be changed to a minimum of 2.4 GB.

Physical memory

Because there are multiple users logging on to the same Windows NT Terminal Server, more physical memory is recommended than for a Windows NT server.

Drives and drive spaces

The default installation of the Integrated Netfinity configures a C: drive of 10 MB, a D: drive of 200 MB and an E: drive of 500 MB. The C: drive contains boot files and should not be altered. However, the default sizes of the D: and E: drives may be too small for most implementations.

The D: drive contains files for the installation CD, the standard size of 200 MB is suitable for Windows NT 4.0, but it may need to be changed for different versions of Windows.

The E: drive contains the actual Windows Terminal server program files and is also where many applications prefer to install files. The initial swap file is also stored on the E: drive. Much of the default 500 MB is taken up by the Windows Terminal Server program files. If you plan on installing other software applications to the same drive as Windows NT or using the E: drive to hold the swap file, the size of this drive should be increased at the time of the installation. The current recommendation is to create an initial drive size of at least 1 GB and install most other applications and the swap file to a separate drive.

Any drive larger than 2048 MB must be formatted as Windows NT File System (NTFS). Even on smaller drives, the NTFS format can provide greater performance and increased security. Since multiple users are logging on to the Windows Terminal Server, security becomes a much bigger consideration. We recommend that you format the E: and subsequent drives as NTFS instead of File Allocation Tables (FAT). This section does not cover the differences between the two drive formats. For more detailed information, please reference the Microsoft documentation.

Due to the Integrated Netfinity's use of the AS/400 disk subsystem, there are no performance gains from using multiple drives or creating Windows NT volume sets. The AS/400 system already spreads data across multiple drives for the best performance. Volume sets can still be used on the Integrated Netfinity servers to group multiple storage spaces into one larger, logical Windows NT drive. Multiple storage spaces and multiple drives under Windows NT can also be used to help organize applications and files.

A good use of a separate storage space is to install a given application to a separate storage space. If an Integrated Netfinity server card fails, that storage space can be linked to another Integrated Netfinity server card, and the application is again available quickly.

To better organize drive space and allow easy increases in size, the swap file can also be placed in a separate storage space.

CD-ROM

Refer to "CD-ROM" on page 521.

18.4 Installation

For detailed steps on installing and configuring the Integrated Netfinity, refer to 9.1.1.1, "Configuring the AS/400 Integrated Netfinity Server" on page 153. A more in-depth discussion can also be found in *AS/400 - Implementing Windows NT Server on the Integrated Netfinity Server*, SG24-2164.

18.5 Startup procedures

You can start and stop Windows NT Server on the Integrated Netfinity Server from either the AS/400 or Windows NT side.

It is important to remember that shutting down the Windows NT Server terminates communications through the Integrated Netfinity Server to the AS/400 system. This is why it may be a good idea to install an additional LAN adapter, outside the control of Integrated Netfinity Server. Then LAN communication with the AS/400 system is not disrupted when the Windows NT server is shut down or restarted.

Starting and shutting down the Windows NT server is described in the following sections.

18.5.1 Starting the Windows NT Server

When you vary on the network server description associated with a Windows NT Server on an Integrated Netfinity Server, the AS/400 system performs a hardware reset on the processor card of the Integrated Netfinity Server. This is the reason why it takes a long time until the startup procedure is complete. You see the big blue IBM logo on the Windows NT Server console attached to the Integrated Netfinity Server, the same as you do on any IBM PC that you purchased recently.

Be aware that the BIOS has been modified. You cannot press F1 to get into the Setup facility or Esc to speed up Power On Self Test (POST). The BIOS boot logic passes control to the boot sector on drive C:, which starts the operating system (OS) loader. You see the typical Windows NT panel, where you can select to boot a Windows NT Server, a Windows NT Server in VGA mode, or PC-DOS. The latter is the DOS mini-boot image that is used during the installation of Windows NT server.

After you select Windows NT server from the menu or the time-out period is over, the Windows NT Server starts, and the logon panel is displayed. Notice that when you vary on the network server, TCP/IP is automatically started, and all TCP/IP interfaces are activated.

To restart Windows NT server on the AS/400 system, you vary the network server description associated with it back on. This is the method you use to restart the server after you shut it down from the AS/400 side. If you shut down a server from the Windows NT server console, you can restart the Windows NT server from either the Windows NT console or from an AS/400 session. To start the Windows NT server from the AS/400 side, first vary off the network server from an AS/400 session and then vary it on.

To vary on the server, perform these steps:

1. Type `WRKCFGSTS *NWS` on an AS/400 command line, and press Enter.

2. Type 1 in the Opt column next to the network server description you want to vary on, and press Enter.

Note

When using the WRKCFGSTS *NWS command interface with multiple Integrated Netfinity Servers, the VRYCFG command processes sequentially. The first Integrated Netfinity Server is varied on completely before the second is started. To speed varying on or off of multiple Integrated Netfinity Servers, a CL program can be written to submit the VRYCFG command to BATCH. This way, they process concurrently instead of sequentially.

The Windows NT server starts. If you have any user storage spaces in user auxiliary storage pools, the AS/400 system automatically mounts that file system (and logs a message to the QSYSOPR message queue).

You should not vary off the network server immediately after varying it on. Wait several minutes to allow the Windows NT Server to start completely. This also allows the service control manager to alert you to any device or service failures.

18.5.2 Shutting down Windows NT server from the AS/400 side

You can shut down the Windows NT server by varying off the network server description for the server on the AS/400 system. If you vary off the network server description, you must vary on the network server description from an AS/400 session to restart Windows NT Server. You cannot restart it from the Windows NT Server console.

Note

When you are running Windows NT server on an Integrated Netfinity Server in your AS/400 system, always shut down Windows NT server before powering down the AS/400 system. If you do not, you risk losing Windows NT server data.

Using the command `PWRDWN SYS *IMMED` does not vary off network server descriptions. The command `PWRDWN SYS *CNTRL D` initiates a vary off, but there is no guarantee that it will complete before the system shuts down. Corruption of the server or network server storage spaces is possible if the AS/400 system powers down before the Windows NT server shuts down completely.

The Restart button on the final Shutdown dialog *cannot* be used to restart the server after it has been shut down from the AS/400 side. At this time, there is no connection between the AS/400 system and Windows NT. Therefore, if you need to restart Windows NT, vary the network server description back on. Notice that if you vary off a network server that has a TCP/IP interface defined on at least one of the Integrated Netfinity Server physical LAN ports, you are forced to reply to the inquiry message, CPA2614 Network server (network-server-name) cannot be varied off at this time. (C G), on the QSYSOPR message queue.

You must respond whether you want the vary off to continue (G) or to be canceled.

To avoid having to respond to this message, use one of the following methods:

- You can end the *external* AS/400 TCP/IP interfaces using the End TCP/IP Interface (`ENDTCPIFC`) command before you vary off the network server description (you do not need to end the interface for the internal LAN).
- Set up a reply list entry to automatically issue a reply to the CPA2614 inquiry message.
- Specify `FRCVRYOFF(*YES)` on the Vary Configuration (`VRYCFG`) command to avoid this inquiry message.

Keep in mind that you need to use one of these options to vary the network server off cleanly if you want to perform an unattended backup of user storage spaces from a CL program.

To vary off Windows NT server, follow these steps:

1. First ensure that the server has no active TCP/IP interfaces:
 - a. Type `NETSTAT` on an AS/400 command line, and press Enter.
 - b. Type `1` to work with TCP/IP interface status.
 - c. Type `10` in the Opt column to change the server TCP/IP interface status to inactive.
2. Type `WRKCFGSTS *NWS` on an AS/400 command line, and press Enter.
3. Type `2` in the Opt column next to the network server description you want to vary off, and press Enter.

The Windows NT server shuts down.

18.5.3 Shutting down Windows NT server from the Windows NT side

You can shut down Windows NT from the Windows NT server console in the same way as you would on a PC-based server, by selecting **Start->Shut Down**. If you shut down from the Windows NT console, you can click the Restart button on the final shutdown dialog box to restart the Windows NT server. The Restart button does not work if you varied off the network server from the AS/400 system.

Shutting down from the Windows NT console shuts down the server, but does not vary off the network server description. At the time when the Restart dialog box is displayed on the Windows NT console, enter the OS/400 command `WRKCFGSTS *NWS`. The Work with Configuration Status - Network Server Status display appears. The network server description is in status SHUTDOWN, and all the other configuration objects are in status FAILED. This is normal.

If you shut down from the Windows NT Server side, you should also restart from the Windows NT console. If you want to restart the Windows NT server from an AS/400 session after shutting down from the Windows NT console, you must first vary off the network server description.

To shut down the Windows NT server from the Windows NT console, perform these steps:

1. Select **Shut Down** from the Windows NT Start menu.
2. Select **Shut down the computer?**

When you install new software on Windows NT server or perform other Windows NT server-side maintenance, you may need to shut down and then immediately restart the Windows NT server. In these cases, select **Restart the computer?** instead of Shut down the computer?.

3. Depending on your OS/400 release level, you may see a shut down confirmation window with the message: Shutting down will stop the external host LAN, and will also end AS/400 communications. Continue?. If you are using the Integrated Netfinity Server LAN ports to connect through to the AS/400 system and do not want to lose the connection, select **No** to stop the shutdown. Select **Cancel** if you do not want this confirmation message to be displayed again. Otherwise, click **Yes**.

Chapter 19. Data replication and data migration

This chapter describes data replication and data migration, along with some of the issues and considerations for choosing to replicate or migrate your data.

19.1 Data replication

Data replication is the process of replicating (copying) database tables to workstations or servers. One of the objectives of data replication is to place data closer to the processing logic that accesses the data to reduce network traffic and speed up response time. This is especially important in distributed and WAN environments.

OneWorld also supports other deployment strategies that may be used to improve OneWorld performance in distributed and WAN environments. These include Partitioning Application Logic, Hardware Tiers, Windows Terminal Server, and OneWorld Java Server. If you are considering replication to have a “hot standby” system, other alternatives, such as hardware solutions and database mirroring, are more appropriate. Such systems are built for high availability. Some of the issues are different from those for regular data distribution.

19.1.1 Planning a replication strategy

Data replication requires you to thoroughly plan the operations of your configurable network environment. Before data replication strategies are adopted, you must make these fundamental decisions:

- Which “replication engine” to use
- What level of support is required to administer the chosen replication facilities
- What tables should be replicated and where should the replicas reside

19.1.1.1 Which replication engine to use

The first decision usually involves deciding between using J.D. Edward’s built-in OneWorld replication facilities or other tools such as those provided by a third-party vendor. The advantage of using the OneWorld replication tool is that it is integrated into the OneWorld product. Third-party replication tools generally offer more functionality than OneWorld.

OneWorld’s data replication facility is suitable for the most basic replication needs. In general, OneWorld data replication is best used when replication needs involve data that is relatively static and the number or size of tables being replicated is relatively small. You should also be aware that changes made using WorldSoftware are not propagated through OneWorld publish and subscribe.

19.1.1.2 Level of support required

The support issue is important regardless of whose data replication facilities are used. Allocating adequate administrative resources to monitor and oversee data replication in any production environment is critical. The integrity of replicated data must be preserved, and error recovery methodologies must be understood and mastered.

19.1.1.3 What tables to replicate and where should they reside

Deciding which tables to replicate and where the replicas should reside is a decision based largely on functional and performance reasons. For example, some applications may access multiple database tables during processing. Splitting them across multiple locations will result in decreased performance. The *OneWorld System Administration* manual contains information on which tables are candidates for replication. However, each customer situation is different. What is a static table to one customer may be a dynamic table to another customer based on their business needs.

19.1.2 Data replication guidelines

When implementing data replication for your environment, follow these guidelines:

- The simplest is the best. The less that can go wrong in replication processes, the less that will go wrong. Over complexity leads to risk due to technical or administrative failure. Therefore, only replicate what is necessary and when it is necessary. Use batch copy routines whenever possible (as opposed to record-based real time).
- Replication uses network bandwidth. Therefore, choose a method and a time that takes advantage of network availability.
- Only perform top-down, one-way replication to ensure that databases remain in synchronization.
- Replicated data should be write-protected to avoid unsynchronized or unintentional changes.

Some tables must only be replicated at certain times. Therefore, establish well-defined operational procedures for what must be replicated, when replication should be performed, and how to check on the success or failure of the process.

19.1.3 IBM and third-party data replication solutions

Non-OneWorld data replication solutions generally provide a greater degree of manageability and reliability in crisis situations. The IBM data replication solution, as well as solutions from Vision Solutions, Inc., DataMirror, and Lakeview Technology, are described in this section. For additional information on data replication solutions from IBM and these vendors, visit the following Web sites:

- IBM: <http://www.software.ibm.com/data/dbtools/datarepl.html>
- Vision Solutions, Inc.: <http://www.visionsolutions.com>
- Lakeview Technology Inc.: <http://www.lakeviewtech.com>
- DataMirror Corporation: <http://www.datamirror.com>

19.1.3.1 IBM DataPropagator Relational 5.1

IBM's DataPropagator Relational 5.1 provides read-only, update-anywhere, and On-Demand replication between relational sources and targets. It defines the architecture for the comprehensive Data Replication Solution.

The DataPropagator Relational consists of the following autonomous components and programs:

- **Administration**

- Control Center is included in DB2 Connect Personal Edition.
- Command line commands can be used to create DataPropagator Version 1 registrations and submissions. The Control Center or DPRTOOLS must be used for administration of DataPropagator Relational 5.1 for AS/400.

- **Capture:** Captures changes made to data on replication sources.

- **Apply:** Reads previously captured, changed data and applies it to target tables.

The Administration component for the DataPropagator Relational 5.1 is called the *Control Center*. The Control Center is the integrated administration tool for DB2 Universal Database V5 and includes replication administration. The Control Center runs on OS/2, Windows NT, Windows 95, and Windows 98 platforms and does not require a local DB2 database. The Control Center is included in the DB2 Connect Personal Edition box. A single user license of DB2 Connect Personal Edition is shipped with DataPropagator Relational 5.1 for AS/400.

The Control Center is used to:

- Define tables as sources, called *replication sources*
- Define views and joins views as replication sources
- Define target table definitions, called *replication subscriptions*
- Clone replication subscriptions to other servers
- Remove replication sources or subscriptions no longer needed

The Capture component captures changes made to data in tables defined as replication sources by reading the database log or journal, without any changes and asynchronously to business applications. The captured changes are placed in staging tables.

DataPropagator Relational Capture for AS/400 supports the remote journal function, which is available on OS/400 V4R2 and V4R3. This function can significantly reduce the CPU and DASD consumption on the primary production system. It does this by making it possible to offload the data capturing process to a different system. Capture for AS/400 also supports the ALIAS feature in SQL, which is available for V4R3 of OS/400.

The Apply component reads the changed data, that was previously captured and stored in a staging table. Then, the component applies it to the target tables. Apply components can also read data directly from source tables, for example, for a full refresh. Supporting update and refresh copying provides greater flexibility and automation in a replication environment.

The Apply component also enhances the data to your specifications as it copies data to the targets. You can exploit the full power of SQL to:

- Create new columns
- Summarize data
- Translate data
- Join data

The Apply component allows you to create:

- Read-only copies
- Updatable copies

The other products in the IBM replication solution, DataPropagator NonRelational, DaraRefresher, DataJoiner, and Lotus Notes Pump, when used in conjunction with DataPropagator Relational, provide a comprehensive cross-platform, cross-database solution for replicating data among many IBM and non-IBM platforms and databases.

19.1.3.2 Vision Solutions

Vision Solutions, Inc.'s Vision Object Mirroring System/400 (OMS/400) is a module in the Vision Suite that performs data replication, which enables the distribution of tasks to remote systems. Vision OMS/400 provides real-time replication using sophisticated communication links and incorporates such features as self-healing techniques, production library monitoring and tuning tools, built-in object refresh capabilities, no file or object limitations, and multi-directional replication. It works with any commercial application, and performs impressively in a high volume, high transaction environment.

The Vision OMS/400 system has a number of powerful features:

- **Self-healing:** Automatically repairs or corrects abnormal conditions, such as communication, synchronization, and system failure recoveries.
- **Group Simulcasting:** Synchronize enterprise-wide data by simulcasting changes from a source system to more than 9,000 target destinations.
- **Virtual Links:** Create almost unlimited prioritized AS/400 links between systems.
- **User Spaces:** Replication process streamlined by taking advantage of user space technology.
- **Self-contained:** Intelligent monitoring process analyzes events and uses self-healing techniques or alerts appropriate personnel.
- **Real-time replication:** The secondary system maintains an exact duplicate of all database files, data areas, and data queues.
- **Synchronization checks:** Optional ongoing validity checking ensures data integrity.
- **Security:** OS/400 protection as well as customized access to product features by profile, group, or user class.
- **Intelligent restart:** Automatic restart after any system termination driven using a pointer system.
- **Transaction filtering:** Automatically filters unwanted entries.
- **Remote Unit of Work Capability:** OMS/400 can execute programs or commands on a remote system.
- **Performance reports:** Throughput optimization using monitoring and tuning reports.
- **API and MI routines:** Allow fast throughput.
- **Dynamic data/object change capture:** Captures without custom commands or re-compiles.
- **Tape offload:** Downloads transactions to tape as needed for total data protection.
- **Languages:** Written in ILE/C and CPI-C code for greater portability and performance.

- **OptiConnect/400 support:** Fully supports the IBM OptiConnect/400 system when a fiber optic bus-to-bus connection is available.
- **Built-in support:** Electronic customer support enables users to access technical support 24 hours a day, 7 days a week, anywhere in the world.

19.1.3.3 DataMirror

DataMirror's *Transformation Server* is a flexible, scalable transformation and replication solution. It offers bi-directional, field-level replication with support for like and heterogeneous platforms and databases, advanced selection and filtering, transformational data enhancement, and cleansing, as well as real-time replication if required.

DataMirror Transformation Server includes the following features:

- **Built-in data transformation:** Enables users to apply data transformation and cleansing routines during replication. Users can translate values, derive new calculated fields, join tables at the source, reformat field sizes, field names and data types, and convert date fields.
- **Open design:** User exits and stored procedures provide the ability to perform record-level transformation through custom programs. Users can integrate processing routines, manipulate files and tables, or pull in data to create derived tables.
- **Mapping-based setup:** Administrators use an intuitive user interface to set up, manage and monitor all replication activity, so it requires no programming changes to applications or databases to implement.
- **Net-change replication:** Transformation Server can be configured to replicate only changes in real-time or on a periodic basis to target systems, avoiding redundant data transfer that ties up system resources and increases communication costs.
- **Selective filtering:** Data can be chosen for extraction based on advanced row or column selection. Users can limit access to sensitive information, populate departmental data marts with user-centric information, or select location-specific data for replication to particular sites.
- **Heterogeneous support:** Transformation Server supports interplatform and interdatabase replication between OS/390, AS/400, Sequent/NUMA-Q, UNIX, Linux, Oracle, Sybase, Pointbase, and Microsoft SQL Server.
- **Flat file replication:** Structured files on UNIX-based, S390, and AS/400 platforms can be exported into a relational format for client server and e-commerce environments.
- **Enterprise administration:** A Java-based graphical user interface is available for configuring and managing replication services across all supported systems and databases, providing a single unified point of administration.
- **OneWorld relational data mart templates:** QuickMarts is an available enhancement to Transformation Server that is a set of pre-configured business intelligence templates that automate the process of transforming source OneWorld data into usable relational data marts.

19.1.3.4 Lakeview Technology

OmniEnterprise is a suite of software products provided by Lakeview Technology for the replication and sharing of data from the AS/400 system and other major DBMSs.

The *OmniEnterprise* data movement methods include change-based data replication, snapshot copying, and bulk extraction and loading. Each method is controlled through a single graphical command console.

OmniDirector is the graphical management console for controlling enterprise-wide data flow. With *OmniDirector*, users design an Enterprise Information Sharing Model (EISM). Administrators use this control center to select data locations, DBMS types, tables, columns, attributes and more. Flexible data partitioning and numerous routing patterns are supported. *OmniDirector* enables automated monitoring of all movement processes through an intuitive graphical display.

OmniReplicator supports heterogeneous change-based replication asynchronously on a scheduled, continuous or event-driven basis. *OmniReplicator* can, with no application alterations, replicate data between different table structures. It also provides multi-directional collision detection and resolution.

OmniCopy enables “snapshots” of single tables, sets of tables, or all source node tables to be copied to the desired targets in full table refresh, row refresh, table merge, or append modes. *OmniCopy* is ideal for automating the periodic distribution of data or refresh of remote systems.

OmniLoader automates cross-DBMS extract and load processes, enabling synchronization of large tables across an enterprise. *OmniLoader* automatically generates utility scripts for the movement of these tables based on the Enterprise Information Sharing Model.

19.2 Data migration

Data migration differs from data replication in that it is usually a one-time task. Data migration can also involve importing data from a totally different application. Normally, if the data is coming from a different system it is reformatted and converted a field at a time on the system that the data is converted from. The major reason for this is that the user is typically more familiar with the export system and can write programs to build tables from multiple sources. If reformatting of the data is required, the task is much more difficult than if reformatting is not required.

The need for data migration frequently involves the loading of data into OneWorld from a flat file. It also can involve loading of data from another database. The source or the target of data migration can be any one of the following options:

- AS/400 DB2
- Microsoft SQL Server
- Oracle
- Microsoft Access
- Flat files

19.2.1 Data integrity

The primary requirement for any tool is that the integrity of the data must be preserved. Potential areas that have caused problems in migrating data to OneWorld include:

- **Leading blanks:** Must not be dropped
- **Binary data within a BLOB field:** Must not be converted
- **Null fields:** Need to be accommodated
- **Negative fields:** Must be preserved
- **Delimiter character ambiguity:** Characters must be unique
- **Unique key violations:** Need to be resolved
- **OneWorld table definition does not exist:** Cause needs to be determined

19.2.1.1 Leading blanks

The J.D. Edwards database has a requirement that significant blanks can exist anywhere within a character data field. This originally came from the S/36 where the data was right justified in character fields. For compatibility reasons, this data format is sometimes still used. An example of this is the business unit field that is in many tables.

OneWorld operates in an ANSI DEFAULT-enabled environment on a Windows NT SQL server. OneWorld operates in this environment *all the time*. One of the ANSI defaults that is enabled when OneWorld runs is the padding of database values with spaces. Thus, when data is added or updated in the table, SQL Server 6.5 appropriately pads it.

When SQL Server 6.5 BCP is used to copy data into tables, it strips off all padding regardless of whether the original data was padded. As a result, when SQL Server 6.5 BCP is used to populate OneWorld tables, it leads to data inconsistency since the table can contain both padded and unpadded data. In addition, Microsoft SQL Server 6.5 contains a bug that occurs when table indexes are searched. The bug is that SQL Server 6.5 ignores the padding (even if a record is stored with padding) when performing searches, therefore, in a table that contains both padded and unpadded data, the results are unpredictable. SQL Server 6.5 may fail to find records that exist in a table. Microsoft says that the results are unpredictable especially in cases that involve a composite index. There are no fixes to SQL 6.5 BCP. The bugs have been fixed in SQL Server 7.0.

The AS/400 CPYFRMIMPF command (Parallel Data Loader) also drops leading blanks unless the character fields are delimited with quote marks. To enable this functionality to work, a new PTF is required.

19.2.1.2 BLOB data

Tables that contain both character data fields, which need to be translated from ASCII to EBCDIC, and BLOB data such as images, which must not be translated, are the most difficult to migrate.

19.2.1.3 Null fields

Many of the migration tools are not able to handle null data. For example, only the fixed format of the Parallel Data Loader allows for a fixed position to be defined to denote a field that contains blanks or nulls.

19.2.1.4 Negative fields

Negative fields have traditionally been a problem in data migration. This is only a potential problem when data is moved from a database to an intermediate format. If the original data source is used, that database also knows the definition of negative fields. If an intermediate format is used, a string format should be used with a separate leading or trailing sign.

19.2.1.5 Delimiter character ambiguity

Another cause of data corruption is if the user chooses a delimiter that also appears as valid data in the data stream. This usually occurs when a special character is chosen instead of using the tab character as a delimiter. If the data is mostly character in format, this can be very difficult to find. Usually the error only shows up as either data shifted from one field to another and a message in a log file that data truncation occurred. Since data truncation is frequently expected, this error is often overlooked. The choice of a comma as a separator between fields is typically not good since a comma sometimes occurs in company names. The choice of a tab character is much better as a field delimiter. The use of fixed format data eliminates this potential problem altogether but at the cost of an increased amount of work.

19.2.1.6 Unique key or data format violations

Unique key or data format violations usually occur when the tables and indexes are defined in one release and then the data is moved to a new definition in a different release of OneWorld. Another known reason for unique key violations is that the data came originally from WorldSoftware, which did not enforce the unique key requirement as rigidly. Normally the duplicate data is a result of a job that abnormally ended in WorldSoftware, and the data simply needs to be deleted.

19.2.1.7 OneWorld table definition does not exist

When you attempt to move all tables from one machine implementation of OneWorld, an error occurs sometimes that states that the OneWorld table definition does not exist. This is not always a legitimate error since a few OneWorld tables are machine dependant. Also, occasionally a table is put in for the next release in one platform and not another.

19.2.2 Data migration methodologies

The primary requirement of any methodology used is that the integrity of the data not be corrupted in the process of migration. Take care to choose a methodology that does not violate the J.D. Edwards database requirement that significant blanks can exist anywhere within a character data field. You must also take care to maintain the integrity of the binary data within a BLOB data type by ensuring that it is not converted or translated.

Some data migration methodologies require that each table be defined to the utility. This requirement normally means that the characteristics of each field must be specified in the format that the loader program desires. For a few tables, this is relatively easy to accomplish. However, if an entire OneWorld database is to be migrated, this means that approximately 1,500 table definitions must be defined on a field-by-field basis.

The choice of the methodology for data migration becomes a trade off between:

- Data integrity
- Speed
- Ease of implementation

The OneWorld data copying application Table Conversion (or CPYTBL for earlier versions of OneWorld) does not cause data corruption and is easy to implement since it uses the existing field definitions from OneWorld. This application also creates both the table and the associated indexes if this is requested.

Note

The OneWorld data copying application is definitely the recommended method of data migration into, within, and out of OneWorld. The only reason not to use this utility is speed.

For very large databases, normally a combination of methods is used to accomplish data migration. The OneWorld Table Conversion is used to create all of the table definitions and to migrate the majority of the tables. You should migrate the tables containing a mixture of BLOB data and normal data using only the OneWorld data copying application. Other very large tables can be migrated using a different methodology to speed up the overall process.

Most methods that gain significant time in migrating the data do so by not maintaining the indexes during the data migration process. The indexes are deleted before the process is started and rebuilt after the data has been migrated. This is less critical if the destination of the data migration is an AS/400 system since the AS/400 system automatically drops maintaining the index on the fly if the number of records added is greater than 10% of the original data size. The other primary source of time saved is that FTP is used to move the data between machines instead of an ODBC (or OCI) connection.

The primary tools to help with the migration of data into OneWorld from other platforms or software are:

- Table Conversion Utility from J.D. Edwards
- Parallel Data Loader from IBM
- Bulk Copy Program (BCP) from Microsoft
- Data Transformation Services from Microsoft
- Oracle Loader from Oracle

19.2.2.1 Table conversion utility from J.D. Edwards

Table conversions are a type of batch process that allow you to perform high-speed manipulations of data in tables. The table conversion tool includes four conversion types, which allow you to do a variety of data manipulations:

- **Data Conversion:** Allows you to transfer or copy data from an input table or business view into one or more output tables using any amount of logic necessary to perform the transfer. You can also use Data Conversion to update records in a table or business view.
- **Data Copy:** Allows you to copy one or more tables from one data source or environment to another data source or environment.

- **Data Copy With Table Input:** Allows you to copy tables based on information from the input table. For example, the input table may provide information about which tables should be copied, where they should be copied, and so on.
- **Batch Delete:** Allows you to delete records from a table or business view.

The Table Conversion tool can use any OneWorld tables or business views or any tables that are not OneWorld tables but reside in a database supported by OneWorld, such as Oracle, Access, DB2 UDB for AS/400, or SQL server. These non-OneWorld tables are commonly referred to as *foreign tables*.

When you create a table conversion, you set up the conversion (which can be saved and run multiple times), review it, and then run it. You can test the conversion first by running it in proof mode, if necessary.

Like reports, table conversions consist of a template and one or more versions. You can override certain properties within a version at run time.

The Table Conversion tool allows you to access any available environment, both for input and output. The environments you choose determine which tables and business views are available for the conversion and where the tables reside. The specification or description of the tables and business views is also determined by the environments you choose.

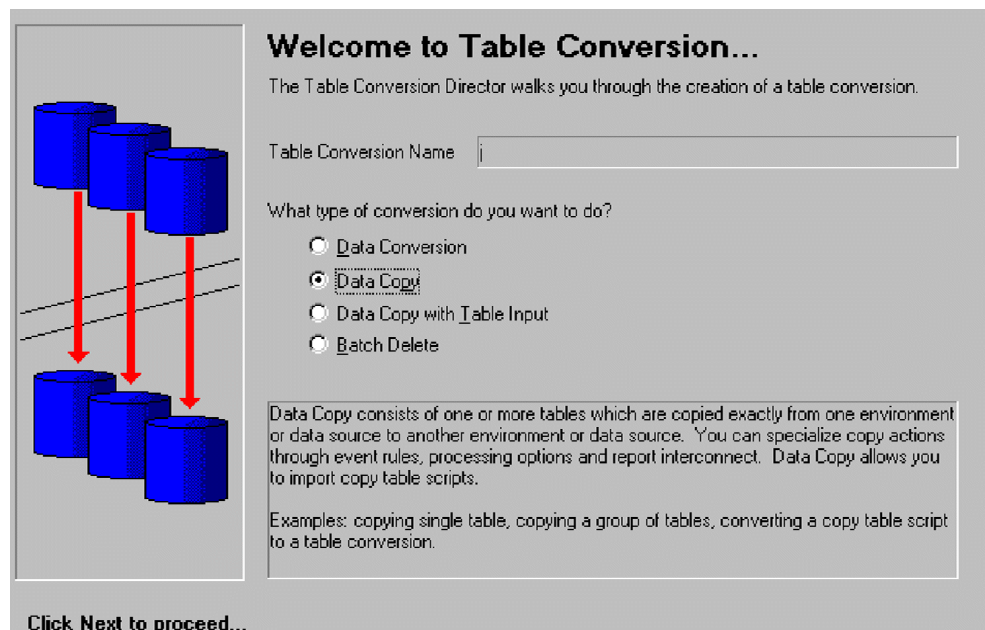


Figure 267. Sample display from Table Conversion

19.2.2.2 Parallel Data Loader

The Parallel Data Loader is the latest parallel database technology added to the DB2 UDB for AS/400. The data loader makes it much easier to quickly load data (especially text delimited files) imported from other Database Servers into AS/400 database files. AS/400 data can also be shared over servers by using the new loader facility to export database files.

AS/400 TCP/IP support made it possible to transfer dumps of database tables from other services onto the AS/400 system. However, if the transferred file

contained fixed format or delimited data, that data was difficult to load into a DB2 UDB for AS/400 table. To complete the load, the system administrator was required to write a custom application program that interpreted the data in these flat files and copied the data into the DB2 UDB for AS/400 database file. The Parallel Data Loader solves this difficulty.

Two new CL commands, Copy from Import File (CPYFRMIMPF) and Copy to Import File (CPYTOIMPF), serve as the interface for the new data loader.

The CPYFRMIMPF command has parameters that allow the layout of the import file to be specified. DB2 UDB for AS/400 uses this information to interpret the source data and loads the data into the target database table. For input files that contain delimited data, the system administrator may specify the delimiters used to separate columns and rows.

The following example shows one possible request for loading a tab delimited import file:

```
CPYFRMIMPF FROMFILE (IMPORT/F4201) TOFILE (PRODDTA/F4201) MBROPT (*REPLACE)
DTAFMT (*DLM) STRDLM (' ') FLDDLM (*TAB)
```

The options on the above import of data are the ones to typically use with OneWorld. OneWorld data typically uses commas and slashes within the data. The disadvantage of the above delimited format is that the NULL function is lost. To export this same data, the following example could be used. Note that the option of *TAB does not exist on the export but it does exist on the import. To accomplish the tab delimited format, hex'05' must be keyed.

```
CPYTOIMPF FROMFILE (PRODDTA/F4201) TOFILE (EXPORT/F4201) MBROPT (*REPLACE)
DTAFMT (*DLM) STRDLM (' ') FLDDLM (X'05')
```

If the import file data is fixed format, without any delimiters, a field definition file must be created to define the layout and format of the import file. The field definition file allows the system administrator to define where data for a column starts and ends, as well as the position of null value indicators for a column.

The following example contains a field definition file that references that file on the CPYFRMIMP command ("- " indicates a comment line):

```
- First column is starting position of field
- Second column is ending position of field
- Third column is position of null value (is any)
```

```
Field1 1 12 13
Field2 14 19 0
Field3 20 30 0
Field4 31 32 33
*END
```

```
CPYFRMIMPF FROMFILE (TESTLIB/FIXEDIMP) TOFILE (TESTLIB/TABLE2) DATFMT (*FIXED)
FLDDFNFILE (TESTLIB/TAB2FLDS) DATFMT (*MDY)
```

If DB2 SMP licensed features are installed, the load operation can be accelerated with parallel processing. This is useful if large import files are to be loaded into DB2 UDB for AS/400 with multiple processors. Once parallelism is activated, DB2 UDB for AS/400 uses multiple tasks rather than a single task to load import files.

A data exporter is also offered in conjunction with the new data loader function. DB2 UDB for AS/400 data can now be exported with the CPYTOIMPF CL command, which operates similarly to the CPYFRMIMPF CL command.

Unfortunately, the data export command cannot take advantage of parallel processing.

Without a PTF, in V4R4 and earlier releases, the AS/400 CPYFRMIMPF command (Parallel Data Loader) drops leading blanks unless the character fields are delimited with quote marks.

19.2.2.3 Bulk Copy Program (BCP) from Microsoft

The Bulk Copy Program (BCP), which is part of Microsoft SQL Server, is another choice for moving data from SQL Server to DB2 UDB for AS/400. The BCP program that is part of SQL Server 7.0 must be used or the integrity of the data will be compromised. The version of BCP that is part of SQL Server 6.5 will truncate the leading blanks when used as an import function.

Microsoft SQL 7.0 and BCP gives the user an fast and easy means of creating scripts to export data to tab delimited flat files. These scripts can easily be run to create a directory of all of the tables to export. That directory can be sent by FTP to another machine using a single MPUT command with no user intervention if prompting is turned off. During a trial migration, about 11 GB of data was transferred in five to seven hours of processing time. The flat files created on the AS/400 system can then be read by using the AS/400 CPYFRMIMPF command to complete the migration process.

The following steps are an example of a method to move all tables owned by a user in MS SQL Server to the AS/400 system. This method can be used to move the majority of the OneWorld tables from an environment on SQL Server to an environment on the AS/400 system. The tables owned by users PRODDTA and PRODDTA were moved into the libraries of the same names on the AS/400 system.

On the Windows NT machine containing the SQL Server database, run the following script using interactive SQL to create the BCP output batch file:

```
select "bcp JDE_PRODUCTION." + rtrim (u.name) + "." + rtrim(o.name)
      + " out e:\bcp\proddta\" + o.name + ".bcp -Usa -P -SDENLTBS20 -c"
      from sysobjects o, sysusers u
      where u.name like 'PRODDTA'
            and u.uid = o.uid
            order by o.name
```

Note the following points, where:

- *JDE_PRODUCTION* is the SQL 7 Database name.
- *DENLTBS20* is the server name containing the SQL 7 Database.
- *PRODDTA* is the user name of the requested files.
- *e:\bcp\proddta* is the location for the BCP files (extracted delimited files).
- *sa* is the predefined user of systems administrator.

The output of the previous script is a series of unload (out) BCP statements (see the following example). It outputs one statement for each table in the database with the specified user. Save the SQL output as a text file, and manually edit out the extraneous output at the beginning and the end of the file using Notepad.

```
bcp JDE_PRODUCTION.PRODDTA.F00022 out e:\bcp\proddta\F00022.bcp -Usa -P -SDENLTBS20 -c
bcp JDE_PRODUCTION.PRODDTA.F0006 out e:\bcp\proddta\F0006.bcp -Usa -P -SDENLTBS20 -c
bcp JDE_PRODUCTION.PRODDTA.F0006D out e:\bcp\proddta\F0006D.bcp -Usa -P -SDENLTBS20 -c
bcp JDE_PRODUCTION.PRODDTA.F0006Z1 out e:\bcp\proddta\F0006Z1.bcp -Usa -P -SDENLTBS20 -c
bcp JDE_PRODUCTION.PRODDTA.F0007 out e:\bcp\proddta\F0007.bcp -Usa -P -SDENLTBS20 -c
```



```

bcpl JDE_PRODUCTION.PRODDTA.F0007Z1 out e:\bcp\proddta\F0007Z1.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F0008 out e:\bcp\proddta\F0008.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00085 out e:\bcp\proddta\F00085.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F0008B out e:\bcp\proddta\F0008B.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F0009 out e:\bcp\proddta\F0009.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00090 out e:\bcp\proddta\F00090.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00090D out e:\bcp\proddta\F00090D.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00091 out e:\bcp\proddta\F00091.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00092 out e:\bcp\proddta\F00092.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00093 out e:\bcp\proddta\F00093.bcp -Usa -P -SDENLTBS20 -c
bcpl JDE_PRODUCTION.PRODDTA.F00095 out e:\bcp\proddta\F00095.bcp -Usa -P -SDENLTBS20 -c

```

On the Windows NT machine containing the SQL Server database, run the batch file you created above, using the DOS command prompt. It creates a flat, tab-delimited file for each table listed in the file. Make sure that space is available on the server for the output files. They will be in the form of the table name with a BCP suffix (F0911.BCP for example).

Use FTP to push the BCP files across to the AS/400 system. After signing onto the AS/400 system in the FTP session and setting both current directories, use the `PROMPT` command to shut off the interactive prompting and `MPUT *.BCP` to perform a multiple table transfer of flat files to the AS/400 system. Do not use the `BINARY` command because the character-based flat files need to be converted from ASCII to EBCDIC.

Create the file structures for the target OneWorld release on the AS/400 system. B733 structures were used for the test migration. The file structures have to be created through the OneWorld installation process or moved from another AS/400 system where an installation for the same release of OneWorld was already done.

Use the Parallel Data Loader feature of the AS/400 system to load the OneWorld tables. The Copy From Import File (CPYFRMIMPF) command is used for this process. An AS/400 CL program can be written to automate this process (see the following sample program). This sample program was written to run interactively, and its message handling would need to be changed to allow it to run in batch. It creates a list of all files to be loaded, clears the previous contents, and loads the files from the flat files. It also records errors to a message queue.

```

/*-----*/
/* COLLECT INFORMATION ON ALL FILES IN XXPRODDTA */
/*-----*/

PGM
DCLF FILE(WORKLIB/DSPOBJD)

CHGJOB LOG(2 10 *MSG) LOGCLPGM(*NO)

CHGSYSLIBL LIB(NEWTEST)
MONMSG MSGID(CPF0000)

/*-----*/
/* GATHER A LIST OF ALL FLAT FILES TO BE LOADED */
/*-----*/

DSPOBJD OBJ(IMPORTDTA/*ALL) OBJTYPE(*FILE) +
OUTPUT(*OUTFILE) OUTFILE(WORKLIB/DSPOBJD) +
OUTMBR(*FIRST *REPLACE)

/*-----*/
/* LOOP THROUGH THE FILE LIST AND CLEAR AND LOAD EACH FILE */
/*-----*/

LOOP: RCVF
MONMSG MSGID(CPF0864) EXEC(GOTO CMDLBL(ENDPGM)) /* eof */
IF COND(%SST(&ODOBAT 1 2) *EQ 'PF') THEN(DO)
CLRPFM FILE(XXPRODDTA/&ODOBNM)

```

```

MONMSG      MSGID(CPF0000) EXEC(DO)
SNDDMSG     MSG('Table ' *CAT &ODOBNM +
                *CAT ' not on AS/400') +
                TOMSGQ(IMPORTDTA/ERROR) MSGTYPE(*INQ)

GOTO        CMDLBL(LOOP)
ENDDO
CPYFRMIMPF  FROMFILE(IMPORTDTA/&ODOBNM) +
                TOFILE(XXPRODDTA/&ODOBNM) MBROPT(*ADD) +
                DTAFMT(*DLM) STRDLM(*NONE) FLDDLM(*TAB)

MONMSG      MSGID(CPF0000) EXEC(DO)
SNDDMSG     MSG('Error in table ' *CAT &ODOBNM) +
                TOMSGQ(IMPORTDTA/ERROR) MSGTYPE(*INQ)

ENDDO
ENDDO
GOTO        CMDLBL(LOOP)
ENDPGM:
          DSPMSG      MSGQ(IMPORTDTA/ERROR) OUTPUT(*PRINT)
          ENDPGM

/*-----*/
Where:
WORKLIB is the library that this program is being run from
IMPORTDTA is the library holding the flat files
XXPRODDTA is the library holding the database files to be loaded

```

19.2.2.4 Data Transformation Services from Microsoft

Data Transformation Services from Microsoft is new with Microsoft SQL Server 7.0. It is designed to be an interactive data migration tool. It is capable of moving data from many different data sources. It seems to work quite well with data sources other than the AS/400 system. Since this is a new product, verify whether fixes exist for the intended usage prior to using this product.

19.2.2.5 Oracle Loader from Oracle

Oracle does not provide a utility to use to export data to a flat file as part of its standard offering. This functionality exists from other third-party vendors, but it is relatively expensive. Data Transformation Services from Microsoft has been successfully used to export data to a Windows NT platform. From this Windows NT platform, the data was then moved to the AS/400 system using the above technique. It is a relatively slow process to move the data from Oracle to Windows NT using this technique, and it should be avoided.

Oracle provides an Oracle Loader program to import data from a flat file. The functions of the AS/400 database definitions can greatly ease the creation of the Oracle loader specifications required to load data. These control files can be easily built in an AS/400 library and sent by FTP as a block of tables using the MPUT command. Likewise, the data tables can also be easily sent by FTP as a block of tables using the MPUT command.

An example of an AS/400 command, a CL program, and a RPG program to build the Oracle loader specifications follows:

```

Generate Oracle Loader Specs (ORACLE)

Type choices, press Enter.

Input Library Name . . . . . > PRODCTL      Character value
Output Library Name . . . . . > ORACLECTL   Character value
Character Set . . . . . > WE8ISO8859P1     Character value
Possible choices for above:
  WE8EBCDIC500 . . . . .                   Character value
  WE8ISO8859P1 . . . . .                   Character value

          CMD          PROMPT('Generate Oracle Loader Specs')
          PARM         KWD(LIBIN) TYPE(*CHAR) LEN(10) RTNVAL(*NO) +
                      DFT(JDFDATA) PROMPT('Input Library Name')
          PARM         KWD(LIBOUT) TYPE(*CHAR) LEN(10) RTNVAL(*NO) +
                      PROMPT('Output Library Name')
          PARM         KWD(CSET) TYPE(*CHAR) LEN(12) RTNVAL(*NO) +

```

```

                                DFT(WE8ISO8859P1) PROMPT('Character Set')
                                KWD(TEXT) TYPE(CHOICE1) RTNVAL(*NO) MAX(1) +
                                PROMPT('Possible choices for above')
CHOICE1:  ELEM                    TYPE(*CHAR) LEN(1) DSPINPUT(*NO) +
                                PROMPT('WE8EBCDIC500') /* AIX */
                                ELEM                    TYPE(*CHAR) LEN(1) DSPINPUT(*NO) +
                                PROMPT('WE8ISO8859P1') /* NT */

```

Sample CL program tied to the above command

```

PGM          PARM(&LIBIN &LIBOUT &CSET &PHONY)

DCL          VAR(&LIBIN ) TYPE(*CHAR) LEN(10)
DCL          VAR(&LIBOUT) TYPE(*CHAR) LEN(10)
DCL          VAR(&CSET ) TYPE(*CHAR) LEN(12)
DCL          VAR(&ACSET ) TYPE(*CHAR) LEN(13)
DCL          VAR(&PHONY ) TYPE(*CHAR) LEN(1)

DCLF         FILE(QADSPOBJ)

CHKOBJ      OBJ(QSYS/&LIBIN) OBJTYPE(*LIB)
MONMSG     MSGID(CPF0001) EXEC(GOTO CMDLBL(BADLIBIN))

CHKOBJ      OBJ(QSYS/&LIBOUT) OBJTYPE(*LIB)
MONMSG     MSGID(CPF0001) EXEC(GOTO CMDLBL(BADLIBOUT))

CHGJOB      JOB(*) LOGCLPGM(*YES)

DSPOBJD     OBJ(&LIBIN/*ALL) OBJTYPE(*FILE) +
            OUTPUT(*OUTFILE) OUTFILE(QTEMP/DSPOBJD)
OVRDBF      FILE(QADSPOBJ) TOFILE(QTEMP/DSPOBJD)

LOOP:       RCVF
            MONMSG     MSGID(CPF0000) EXEC(RETURN)

            IF          COND(&ODOBTP = '*FILE' & &ODOBAT = 'PF') +
                        THEN(DO)
                DSPFFD   FILE(&LIBIN/&ODOBNM) OUTPUT(*OUTFILE) +
                        OUTFILE(DAAJDE/DSPFFD) OUTMBR(*FIRST +
                        *REPLACE)
                CRTPF    FILE(&LIBOUT/&ODOBNM) RCDLEN(64)
                MONMSG   MSGID(CPF0000) EXEC(CLRPFM +
                        FILE(&LIBOUT/&ODOBNM))
                OVRDBF   FILE(ORACLE) TOFILE(&LIBOUT/&ODOBNM)
                CHGVAR   VAR(&ACSET) VALUE('A' *CAT &CSET)
                CALL     PGM(ORACLEFFD) PARM(&ACSET)
                ENDDO
            GOTO        CMDLBL(LOOP)

BADLIBIN:   SNDUSRMSG  MSG('Try again! Bad input library.')
            GOTO        ENDPGM

BADLIBOUT:  SNDUSRMSG  MSG('Try again! Bad Output library.')
            GOTO        ENDPGM

ENDPGM:     ENDPGM

```

Sample RPG program ORACLEFFD called from the proceeding CLP

```

F*****
F*
F*   File from DSPFFD
F*
FDSPFFD  IPE E                               DISK
F*-----
F*
F*   Output file of fixed length record
F*
FORACLE  O  F      64                       DISK
F*-----
IQWHRFFD  01
C*****
C*
C          *ENTRY  PLIST
C          PARM           ACSET  13
C*
C          *IN10   IFEQ *OFF
C          MOVE ACSET           CSET  12

```

```

C          MOVE '.DAT'' ' DAT      5
C          CAT  DAT:0  FILE  14
C 01       WHFILE    EXCPTHEAD
C          SETON                                10
C          ENDIF
C* -----
C          WHFOBO    ADD  WHFLDB    END    50
C          END      SUB   1         END
C* -----
C          END      IFNE WHRLEN
C          SELEC
C          WHFLDT    WHEQ 'A'
C          WHVARL    IFEQ 'Y'
C          MOVEVL 'VARCHAR, 'TYPE    8
C          ELSE
C          MOVEVL 'CHAR, 'TYPE
C          ENDIF
C          WHFLDT    WHEQ 'P'
C          MOVEVL 'DECIMAL, 'TYPE
C          WHFLDT    WHEQ 'S'
C          MOVEVL 'ZONED, 'TYPE
C          ENDSL
C          EXCPTWORA
C          ELSE
C          SELEC
C          WHFLDT    WHEQ 'A'
C          MOVEVL 'CHAR 'TYPE
C          WHFLDT    WHEQ 'P'
C          MOVEVL 'DECIMAL 'TYPE
C          WHFLDT    WHEQ 'S'
C          MOVEVL 'ZONED 'TYPE
C          ENDSL
C          EXCPTWORA
C          EXCPTFOOT
C          ENDIF
OORACLE E          WORA
O          WHFLDI    10
O          25 'POSITION('
O          WHFOBO    30
O          31 ':'
O          END      36
O          37 ') '
O          TYPE     46
OORACLE E          HEAD
O          23 'LOAD DATA CHARACTERSET '
O          CSET     + 0
OORACLE E          HEAD
O          8 'INFILE ''
O          FILE     + 0
O          + 1 '"fix'
O          WHRLEN   + 1
O          + 0 ''
OORACLE E          HEAD
O          15 'PRESERVE BLANKS'
OORACLE E          HEAD
O          10 'INTO TABLE'
O          WHFILE   + 1
OORACLE E          HEAD
O          1 '('
OORACLE E          FOOT
O          1 ') '

LOAD DATA CHARACTERSET WE8ISO8859P1
INFILE 'F0084.DAT' "fix 00176"
PRESERVE BLANKS
INTO TABLE F0084
(
MPMNI          POSITION(00001:00010) CHAR,
MPSELN        POSITION(00011:00013) DECIMAL,
MPPHTH        POSITION(00014:00014) CHAR,
MPPTH         POSITION(00015:00134) CHAR,
MPUSER        POSITION(00135:00144) CHAR,
MPPID         POSITION(00145:00154) CHAR,
MPUFMJ        POSITION(00155:00160) ZONED,
MPJOBN        POSITION(00161:00170) CHAR,
MPUPMT        POSITION(00171:00176) ZONED
)

```

Chapter 20. Server-to-server communication

This chapter describes the options available for connecting AS/400 OneWorld servers with other (AS/400 or non-AS/400) OneWorld servers, for example:

- An *AS/400* OneWorld Application Server to an AS/400 OneWorld Database Server
- A *non-AS/400* OneWorld Application Server to an AS/400 OneWorld Database Server

20.1 Ways of connecting AS/400 OneWorld servers

Prior to Release B733 base, OneWorld processed Extended Dynamic SQL statements via a system API called QSQPRCED. This interface only executed statements locally, that is, when the Application Server and Database Server were on the same system. If they were on two different systems, the only OneWorld protocol available at that time was the slower JDENet/JDBNet middleware designed to work between two heterogeneous systems. See 20.5, “Connecting an AS/400 server with a non-AS/400 server” on page 555.

OneWorld B733 was rewritten to use the QBFC APIs, which allows the AS/400 OneWorld Application Server to communicate with an AS/400 OneWorld Database Server using the DBDR, the OneWorld database driver for the AS/400 system. The QBFC APIs, in turn, call the local QSQPRCED APIs. This change enables OneWorld to have three connectivity options, which no longer require JDENet/JDBNet:

- **OptiConnect/400 (OptiConnect):** An AS/400 OneWorld Application Server communicates with an AS/400 OneWorld Database Server over a high-speed fiber-optic bus.
- **Virtual OptiConnect between Logical Partitions (LPARs):** Beginning with V4R4 of OS/400, a single AS/400 system can be divided into multiple, independent logical partitions. Logical Partitions communicate using Virtual OptiConnect.
- **TCP/IP network:** Communications between an AS/400 with OneWorld application serving and another AS/400 system with database serving occurs over a TCP/IP LAN or WAN, using the OneWorld DBDR driver.

20.2 OptiConnect/400 and virtual OptiConnect

Both OptiConnect/400 and virtual OptiConnect between LPARs are high-speed connections. The throughput of the OptiConnect link is 1063 Mbps, while the virtual OptiConnect can achieve similar or better throughput rates.

20.2.1 Benefits

Both OptiConnect/400 and virtual OptiConnect offer you the following advantages for communications:

- **High performance:** This is due to fast data transfer and dedicated link between the two systems.

- **Security:** Server-to-server traffic has exclusive use of the connection, unlike an external TCP/IP environment where other traffic may be sharing the wire.
- **Horizontal growth and flexibility:** The ability to add up to a total of 14 AS/400 systems in one OptiConnect cluster.
- **Reliability:** Link redundancy is standard with two sets of fiber optic cables between hub and satellite. Path redundancy is optional where you can configure two hubs, both of which are recognized and used by OS/400.

20.2.2 OptiConnect/400

This section describes the use of high speed fiber-optic links in a OneWorld implementation. OptiConnect/400 is a combination of hardware and software that allows you to connect multiple high-end AS/400 systems. This connection does not use a network such as LAN, ISDN, or FDDI. Rather, each AS/400 system in an OptiConnect network is connected to another system with a dedicated link. The connection may be best viewed as a device connection where one system accesses another system as if it is an attached device.

OptiConnect/400 hardware consists of OptiConnect Receiver cards that are installed in a dedicated I/O Expansion Tower and are connected to another AS/400 system through fiber optic cables. OptiConnect/400 hardware is limited to 500 meters over the 1063 Mbps (megabit per second) link used on AS/400 RISC models.

OptiConnect for AS/400 software consists of:

- A set of APIs that interfaces to the OptiConnect/400 optical bus
- Additions to OS/400 that provide fast path Distributed Data Management (DDM) or Distributed Relational Database Architecture (DRDA) access across an optical bus
- A connection manager that manages OptiConnect resources
- An agent job that runs on the server on behalf of client requests
- Additional AS/400 objects that support and control the connection manager and agent jobs

Note

Information-only RPQ 843871 describes OptiConnect/400 in greater detail. Please talk to your IBM marketing representative for more information.

20.2.2.1 OptiMover for OS/400

OptiMover for OS/400 is a special, low-cost PRPQ version of OptiConnect for OS/400 software (V3R6 or higher). It enables non-DDM functions to use the OptiConnect high-speed link, allowing system bus-level connections between AS/400 systems. Business Partner software packages written to the OptiMover/400 APIs allow customers to take advantage of this lower priced option. Since OptiMover/400 costs less and includes the APIs that OneWorld needs, we use this software in the following discussion. Of course, OptiConnect software would work as well.

Note

OptiMover, 5799-FWQ, is described in greater detail in PRPQ 84291.

20.2.2.2 Uses for OptiConnect/400

OptiConnect/400 allows AS/400 applications to perform inter-system database access across a high-performance interface. There are two primary uses for the OptiConnect technology:

- *Horizontal growth* is the traditional and most popular use. By separating database operations from application workload, multiple systems can operate as a cluster to grow computing power beyond what a single system can provide.
- *High availability* is the second use for OptiConnect technology. When used in conjunction with business partner applications, such as those offered by DataMirror, Lakeview Technology and Vision Solutions, efficient high availability solutions (as well as data replication) can be achieved. The OptiConnect/400 technology provides the most efficient use of CPU resources to allow fast replication of data between systems.

OneWorld's use of OptiConnect in this chapter is based on horizontal growth.

20.2.2.3 OptiConnect clusters

An OptiConnect cluster consists of a collection of systems, each of which dedicates a system bus to connect to a common or shared bus. The system that provides this shared bus is referred to as the OptiConnect *hub* system. The systems that attach to this shared bus are referred to as OptiConnect *satellite* systems. Figure 268 on page 548 illustrates a network consisting of one hub system (hub machine) and two satellite systems (satellite machines).

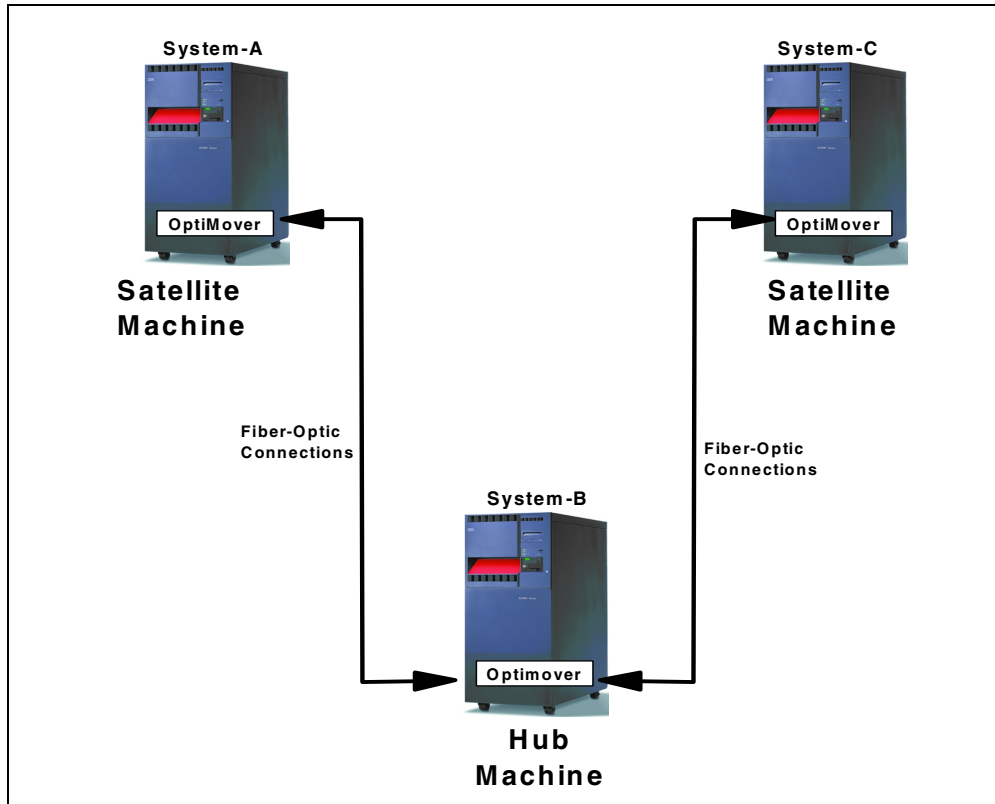


Figure 268. A single path network

The hub machine provides the communication route for all machines on the network. This includes communications between the hub and a satellite, as well as communications between two satellites. There is no processing overhead of any significance on the hub machine. The OptiConnect cluster comes standard with *link redundancy*, where there is a backup cable connecting the OptiConnect cards in case the primary cable fails. If the hub machine fails for any reason, all communications routed through this hub also fail. Consequently, the network can fail.

A dual path configuration (*path redundancy*) is also available. This introduces a redundant hub into the network so that if one hub fails, the redundant hub can take over and the network remains operational. Figure 269 illustrates a network consisting of two hub machines and two satellite machines.

Optionally, in addition to or instead of the dual path configuration, the TCP/IP network on your Opticonnected systems can be used as an alternate path. OneWorld was written to look for an OptiConnect connection first and a TCP/IP connection second, should OptiConnect between hub and satellite fail for any reason. At B733.2, a OneWorld user would receive a communications failure and then have to exit OneWorld and re-signon. However, the OneWorld kernels would continue to communicate with the Database Server over the TCP/IP network.

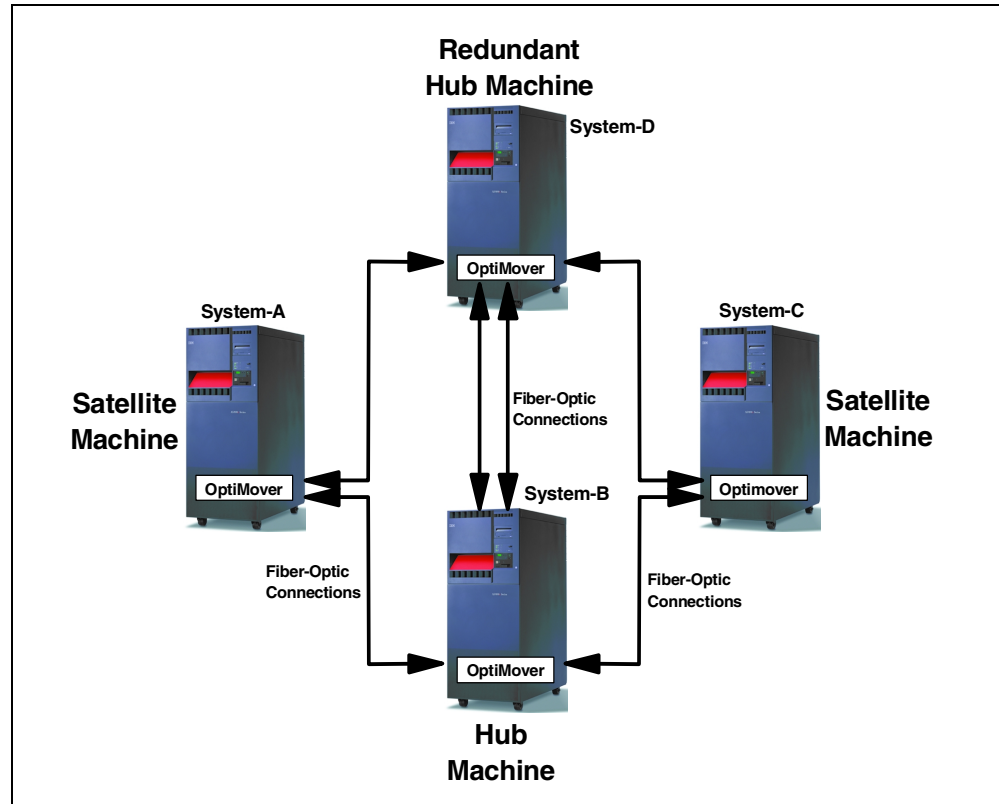


Figure 269. A dual path network

An OptiConnect cluster can consist of up to 14 systems (one hub and 13 satellites), with full system-to-system connectivity and up to 32 systems in complex structures. Interoperability between OS/400 versions is maintained so that systems at different release levels can be connected in the same cluster.

Using OptiConnect clusters with OneWorld

OptiMover/400 support does not enforce any restrictions as to where OneWorld database or Application Servers should be located. However, given the importance of the hub machine in the communications network and the critical nature of the OneWorld Database Server, the OptiMover/400 hub and OneWorld Database Server are best located on the same machine. The OneWorld Application Servers are located on the satellite machines.

If a satellite machine fails, only the corresponding Application Server is impacted. The failure of the hub results in both communications and the OneWorld database becoming unavailable. This arrangement reduces the exposure of the total system to a single point of failure only and maximizes the availability. The system has the potential to be available as long as the hub machine is up.

The arrangement of database and Application Servers is illustrated in Figure 270 on page 550.

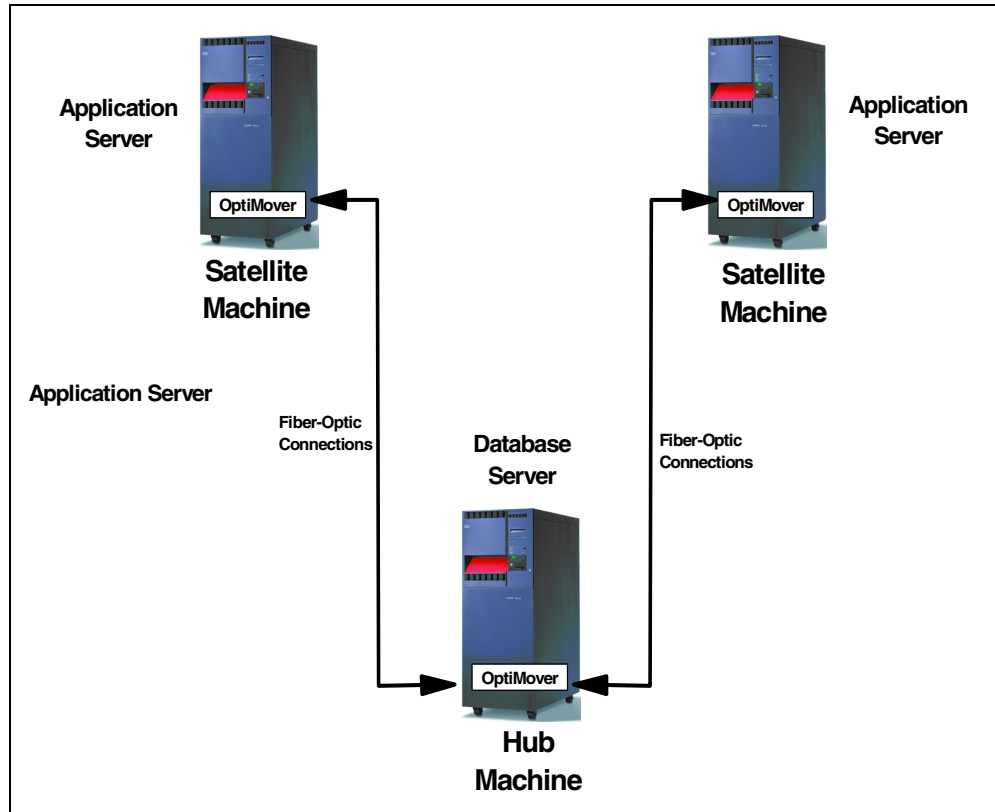


Figure 270. OneWorld server placement

On the other hand, if the Database Server is placed on a satellite machine, the availability of the system is impacted if the satellite (and the associated Database Server) or if the OptiMover/400 hub machine fails.

20.2.3 Virtual OptiConnect

Virtual OptiConnect provides a hardware path to communications software that connects a logical partition with another partition. While this path uses TCP/IP, it does not involve any communication input/output processors (IOPs). Data is transferred directly from a disk in one partition to a disk in another partition using Direct Memory Access (DMA) function, which provides high data throughput between partitions. DMA is not directly available to user programs. The user programs have to use TCP/IP or SNA functions.

The benefits of using Virtual OptiConnect in a OneWorld environment include:

- **Keeping production and development separate:** You may want to have your production and development environments in different LPARs and use virtual OptiConnect for object transfer to update production code, for example.
- **Different workloads:** There may be times where you want to run other applications on separate systems such as HTTP serving, data warehousing, UBEs, or even set up two different Application Servers to talk to a single Database Server.

For additional information, requirements, and instructions on virtual OptiConnect and how to create logical partitions, refer to *Slicing the AS/400 with Logical Partitioning: A How to Guide*, SG24-5439.

20.3 Network

Since OneWorld implementation requires a TCP/IP network, one benefit of using a network to connect two AS/400 OneWorld servers is that the network is already in place. There is no additional hardware or software expense.

While you can choose to implement either OptiConnect or a network for server-to-server communications, you can also install both. OneWorld is designed to route data transfers first over OptiConnect and then, if it is unavailable, route over the network. Therefore, if the OptiConnect solution should become unavailable for any reason, the network can take over after the OneWorld user would have to exit OneWorld and sign on again. While the throughput may be slower over the network due to high traffic, you would have an extra backup path available.

Note

With OS/400 V4R5, which supports the 1Gbps Ethernet and 100 Mbps Token-Ring, a dedicated LAN connection between two servers may also be a viable option for a primary connection.

20.4 Setting up an AS/400-to-AS/400 connection

This section discusses some of the considerations related to setting up an AS/400-to-AS/400 connection in the OneWorld environment.

20.4.1 OptiConnect considerations

To implement a OneWorld multi-tier solution using OptiConnect, your environment must meet the following minimum requirements:

- Have two or more AS/400 systems.
- All AS/400 systems require OptiConnect hardware and either the OptiMover for AS/400 software PRPQ or OptiConnect for AS/400 software. For the rest of this section, we use OptiMover/400 only in our examples.
- All AS/400s require compatible PTFs.

20.4.1.1 Required PTFs

There are PTFs that must be applied before installing and configuring OneWorld B73.3. The current PTFs can be obtained by following the Informational APAR for your version of OS/400. Please refer to the site at:

<http://www.as400.ibm.com/service/bms/jde-support.htm>

Note

A quick test to see if you have the PTFs installed is to run the `WRKOBJ QUSRSYS/QBFCPRCED` command. If the object is not found, you need PTFs from the Informational APAR.

20.4.1.2 Installing OptiConnect

The OptiConnect hardware and software can be installed in any order. The OptiMover PRPQ needs to be installed on all AS/400 systems as discussed in *OptiMover for AS/400*, SC41-0626.

Note

For OptiMover software, 5799-FWQ, there is a required PTF SF50277. You need to order this PTF (downloadable) separately because it is at the V3R7 product level and this PTF will not be on the current Cumulative PTF package. To verify the PTF, please type:

```
DSPPTF 5799FWQ
```

Press Enter to view the PTFs.

20.4.1.3 Authorities

After you installed all required PTFs, verify object authority to the server program and the service programs that run OneWorld on all AS/400 systems that use OneWorld and OptiConnect. The required authority level is shipped with these six programs. Use the Work with Object Authority command as follows:

```
WRKOBJ OBJ(QUSRSYS/QBFC*)
```

Select option **5** to display and verify the authorities as listed below, and select option **2** to edit them. Instead of option 2, you can use the Grant Object Authority (`GRTOBJAUT`) command to grant proper authorities as follows:

```
GRTOBJAUT OBJ(QUSRSYS/QBFCPRCED) OBJTYPE(*SRVPGM) USER(*PUBLIC) AUT(*USE)
GRTOBJAUT OBJ(QUSRSYS/QBFCGETKEY) OBJTYPE(*PGM) USER(*PUBLIC) AUT(*USE)
GRTOBJAUT OBJ(QUSRSYS/QBFCSETP) OBJTYPE(*PGM) USER(QUSER) AUT(*USE)
GRTOBJAUT OBJ(QUSRSYS/QBFCSETP) OBJTYPE(*PGM) USER(*PUBLIC) AUT(*EXCLUDE)
GRTOBJAUT OBJ(QUSRSYS/QBFCRECV) OBJTYPE(*PGM) USER(*PUBLIC) AUT(*USE)
GRTOBJAUT OBJ(QUSRSYS/QBFCCLISTEN) OBJTYPE(*PGM) USER(*PUBLIC) AUT(*USE)
```

20.4.1.4 User profiles

OneWorld user profiles must have the same job description and password on all systems that utilize the OptiConnect facilities. It cannot have an expired password.

20.4.1.5 Starting the OptiConnect environment

The QSOC subsystem must be started on the OptiConnected machines with the following command:

```
STRSBS QSOC/QSOC
```

You can verify success by checking for messages with the following command:

```
DSPMSG QSYSOPR
```

20.4.1.6 Setup OneWorld OCM mappings

The F98611 table needs to be altered to reflect the remote AS/400 system. The record corresponding to the remote data source fields are:

- **OMSRVR:** The setting in this field needs to be the name of the remote AS/400 system. Prior to the remote AS/400 DB access support, this field was ignored for data source records using the DBDR OMDLLNAME. For example, if you have records where OMDLLNAME is set to DBDR, ensure the name of the server is set correctly. Local data sources should be set to the name of the local machine.
- **OMDLLNAME:** This setting should be DBDR, which is the same as the local AS/400 data source records.

20.4.1.7 OptiMover API Agent

When the QBFConnectDB function is called by OneWorld on the local system with the Application Server, an agent job is started on the Database server in the QSOC subsystem, with a name of APIAnnnnn. This is the OptiConnect version of the TCP/IP listener job, which is started automatically.

Note

OptiMover/400 uses the IPC shared memory address 9999. Avoid assigning the jde.ini file setting, IPCStartKey, a starting value within the range of 9000 to 9999.

20.4.1.8 Verifying the OptiConnect connection

There are two ways to verify the OptiConnect connection, depending on which software is used for OptiConnect:

- If OptiConnect software is used, verify that the OptiConnect has system-to-system connection within the cluster, type the `VFYOPCCNN` command, press Enter. Check for messages on your screen and in your job log. This method verifies the connection by exchanging data between the two systems using DDM.
- If OptiMover or OptiConnect is used, verify your connections with the `QSOC/WRKOPCACT` command. The status of your system and resources should be either varied on or active, if the connection is established.

To verify that data traffic is actually flowing across the OptiConnect link (such as traffic generated by PORTTEST or OneWorld users), observe the number of kilobytes transferring across the link as you continue to refresh the screen by pressing F5.

20.4.2 Considerations in a non-OptiConnect environment

This section covers some of the considerations in a network (non-OptiConnect) environment, where two or more AS/400 system are connected via a TCP/IP LAN or WAN.

20.4.2.1 Required PTFs

Refer to 20.4.1.1, “Required PTFs” on page 551.

20.4.2.2 Setting up TCP/IP

See Appendix B, “TCP/IP basic installation and configuration” on page 781, for assistance in setting up the TCP/IP network.

20.4.2.3 Authorities

Refer to 20.4.1.3, “Authorities” on page 552.

20.4.2.4 User profiles

OneWorld user profiles must have the same job description and password on all systems utilizing the QBFC TCP/IP facilities. It cannot have an expired password.

20.4.2.5 Configuring the remote database connection

On all AS/400 systems, add a service by either using CFGTCP option 21 (Configure related tables) and choosing option 1 (Work with service table entries) to add a new entry. Or, execute a command similar to the following example:

```
ADDSRVTABLE SERVICE('<servicename>') PORT(7777) PROTOCOL('tcp') TEXT('OneWorld  
Remote AS/400 DB access via TCP')
```

Note

The service name and protocol values are case sensitive. The single quotes shown in the command above preserve the value's character case.

An entry should be created in the [DB SYSTEM SETTINGS] section in the INI file to specify the service name for each remote AS/400 system to which you require database access. The format of the setting is:

```
RemoteAS400Name=ServiceName
```

For example, if the name of your remote AS/400 system is *JDE2* and the serviceName you created is *OW_RmtDB*, the setting in the [DB SYSTEM SETTINGS] section is:

```
JDE2=OW_RmtDB
```

20.4.2.6 Starting the listener job

Prior to running OneWorld using the TCP/IP remote AS/400 DB access support, the QBFCLISTEN program needs to be running on the remote system. This should be submitted to the system where the remote database jobs will run by using the SBMJOB command.

QBFCLISTEN program has three parameters:

- The first is the type of connection (right now, the only supported connection type is TCP).
- The second is the service name of the port on which the job should listen. Moreover, you need to ensure the service name is added to the system service name table (see the previously described instructions).
- The last is the library name where the QBFCRECVR program resides.

A typical call to start the listener program is:

```
SBMJOB CMD(CALL PGM(QBFCLISTEN) PARM(TCP '<servicename>' QUSRSYS))
JOB(OW_RMTDB) JOBQ(QSYSNOMAX)
```

Note: You may want to run the QBFCLISTEN program in the QSYSWRK subsystem using JOBQ(QSYSNOMAX) on the SBMJOB command. However, you can submit it to whichever job queue makes sense for your environment.

20.4.2.7 Verifying the TCP/IP installation

On the AS/400 command line, type:

```
NETSTAT
```

Press Enter. Select option **3** (Work with TCP/IP Connection status) and page down until you see as-data in the local port column. Then press F11 and continue to refresh by pressing F5 as you watch the bytes in and bytes out columns continue to increment.

20.5 Connecting an AS/400 server with a non-AS/400 server

Prior to OneWorld Release B733 base, the only protocol for connecting two heterogenous systems (for example, AS/400 with UNIX) running OneWorld servers was JDENet/JDBNet. The reason for that was, OneWorld processed Extended Dynamic SQL statements via a system API called QSQPRCED. QSQPRCED only executed statements locally, that is when the Application Server and Database Server were on the same system. If they were on different systems, JDENet/JDBNet middleware, designed to work between two heterogenous systems, was used.

This section describes how to connect an AS/400 OneWorld server with a non-AS/400 OneWorld server using JDBNet.

For two computers to communicate with each other, they must share a communication protocol or a set of protocols. A communication protocol is a formal set of rules that specifies the format and relationship for exchanging data among different devices. JDENet is the J.D. Edwards proprietary communications middle ware package that provides server-to-server and client-to-server communication.

The JDENet architecture is message based. It uses a message type identifier to route messages to a range of processes dedicated to handling requests that fall within a specified message range. In OneWorld Release B73.3, JDENet is comprised of twelve dedicated processes. Each of these processes and their respective message ranges are defined in the logic server jde.ini file.

For this section, the Type 7 JDBNet Server-to-Server communication process holds our interest. When configured, the JDBNet dedicated kernel process makes it possible for data to be shared between OneWorld supported Relational Database Management Systems (RDBMS). This allows OneWorld servers to initiate requests from other servers. An example of this is server-to-server push replication.

The JDBNet process relies on a number of OneWorld and host configurable components. For more information on JDENet, refer to 4.8.1, "JDENet" on page 66.

20.5.1 Two or more host servers running OneWorld code

Two or more host servers must have OneWorld code installed on them. They can consist of any OneWorld supported platform, for example, two AS/400 systems, an AS/400 and RS/6000 server, an AS/400 and Intel Windows NT server, and an AS/400 and HP9000 server.

It is not necessary for the server map tables to reside on the Enterprise Server. If these tables do not reside on the host server, the host jde.ini file must be modified to use the appropriate JDBNet database driver shown in the following example. This value is specified as the DB SYSTEM SETTINGS section Load Library statement in the Enterprise Server jde.ini file.

[DB SYSTEM SETTINGS]	<- In this example, an AS/400 system is our host.
Server=<host_name>	<- Identify the host name where SVM resides.
Load Library=DBDR	<- Change from DBDR to JDBNET if the server map tables used by this OneWorld AS/400 Enterprise Server resides on another system.

20.5.2 Establishing a OneWorld service port

Sockets provide a duplex communication channel between one or more systems. JDENet uses stream sockets to provide end-to-end communications. OneWorld systems use a common port to establish socket-based system connections. The assigned port is unique to each OneWorld release. The port used by OneWorld Release B7332 for socket connections is 6008.

It is possible to add a service table entry that associates a service name, for example, jde_server, with a port number and protocol. To simplify this process, J.D. Edwards recommends defining this port by number, rather than by service name. If you choose to use a service name, note that all entries referring to the name are case sensitive.

[JDENET]	
serviceNameListen=6008	<- Define the port by number.
serviceNameConnect=6008	<- Define the port by number.

The jde.ini file used by all OneWorld systems consists of two statements in the JDENET section. The OneWorld JDENet/JDBNet process uses the values defined in these statements to establish the port number used by said OneWorld system. It is imperative that all OneWorld systems at the same release level share the same port. This facilitates using socket-based connections between OneWorld systems.

20.5.3 Defining host table entries

Each computer system on your network is known as a host. The host table allows you to associate a host name to an Internet address.

When configuring OneWorld on an AS/400 server, you are directed to use the CFGTCP menu's option 10, Work with TCP/IP host table entries, to add a TCP/IP host table alias entry on your AS/400 system. The added entry should be

host_name.domain_name and associated to the host system's Internet address entry. The values that make up the host_name.domain_name originate from the CFGTCP menu's option 12, Change local domain and host names.

Note

All entries referring to the host name's case (upper, lower, or mixed) should match the previously mentioned host table entry and local domain and host names applications. It should also match the following OneWorld tables and files:

- jde.ini file
- Data Source Master File: All Data Source *Server Name* field (OMSRVR)

The OneWorld JDENet/JDBNet processes query the system and determine the host name and domain name. These two values are concatenated in the manner previously described to establish a socket connection with another OneWorld system.

Ultimately, from a OneWorld client workstation, you must be able to perform a PING on the AS/400 server by name. From an AS/400 server, you must be able to perform a PING on another AS/400 server by name. The IP address returned when performing a PING on these systems must be the same address associated to the system's TCP/IP host table entry for host_name.domain_name. If your AS/400 system has two configured network interface cards (NIC), it is possible that the IP address used by a system is not the same as the IP address associated with the host table host_name.domain_name entry.

In such a case, all OneWorld jde.ini files, including those on client workstations and Enterprise Servers, must be modified. Replace all references to the host server name, for example, Server=JDECS9, with the IP address associated to the host table host_name.domain_name entry. Add the case-sensitive statement NetHostName=<host_srvr_IP_addr> to the JDENET section of all jde.ini files (illustrated here):

```
[JDENET]
NetHostName=10.21.34.58
```

Make the same changes to the Deployment Server x:\oneworld\b7332\OneWorld Client Install\misc\JDE.INI files.

20.5.4 Modifying the Server Data Source Master table

Traditionally the Server Data Source Master table (F98611) is created on each AS/400 Enterprise Server with a DLL Name (OMDLLNAME) of *DBDR*. The DLL Name value that is specified depends on the platform and database used. Valid JDBNet values for specific machines and databases are shown in Table 43.

Table 43. JDBNet DLLs for specific machines and databases

OneWorld host system to RDBMS type	DLL, service program, shared library
AS/400 to DB2/400	DBDR
AS/400 to any other server RDBMS	JDBNET
HP9000 to DB2/400	libjdbnet.sl

OneWorld host system to RDBMS type	DLL, service program, shared library
RS/6000 to DB2/400	libjdbnet.so
Intel to DB2/400	jdbodbc.dll
Alpha to DB2/400	jdbnet.dll

As shown in the Table 43, the DBDR service program (AS/400 DLL equivalent) is used when an AS/400 host system connects to a DB2/400 database on the local machine. When the AS/400 system to any other server RDBMS (excluding another DB2/400 database), it uses the JDBNET DLL driver instead.

When data is distributed across Relational Database Management Systems (RDBMS), the OneWorld client workstation uses third-party database drivers to access the database. More specifically, IBM Client Access/400 and Microsoft SQL Open Database Connectivity (ODBC) and Oracle Connectivity Interface (OCI) drivers are used to establish RDBMS connections. The JDBNet kernel process provides the same functionality for server-to-server RDBMS communications.

20.5.4.1 Modifying the server data sources

Make the following changes to the “Server” (not the System - B7332 DS) Data Source Master table:

1. From a OneWorld client workstation, go to the menu GH9011. Select **Database Data Sources**.
2. Highlight the entry in the grid that corresponds to your <host_name_1> - B7332 Server Map data source. Click **Select**.
3. In the display that follows, change the data source use value from DB to “*”.
4. Tab to the QBE line. In the Server Name column, enter:

```
<> host_name_1
```
5. Click **Find**.
6. Update each data source entry that is displayed by changing the defined DLL name from DBDR to JDBNET.
7. Repeat these steps for your other OneWorld logic servers, for example, *host_name_2*.
8. Check each Data Source definition for:
 - Data Source Name
 - Database Name
 - Server Name

For all DB type (not SVR type) Data Sources, the *Database Name* must be identical to the *Data Source Name*. Also, the *Server Name* field should identify the host name on which the database resides. The OneWorld JDBNet kernel process relies largely on these values.

Table 44. Example of B7332 server map data sources

DATA_SOURCE_NAME	DATABASE_NAME	SERVER	DLL_NAME
AS/400 Common - CRP	AS/400 Common - CRP	HOSTSVR1	DBDR
AS/400 Common - Production	AS/400 Common - Production	HOSTSVR1	DBDR

DATA_SOURCE_NAME	DATABASE_NAME	SERVER	DLL_NAME
Business Data - CRP	Business Data - CRP	HOSTSVR1	DBDR
Business Data - PROD	Business Data - PROD	HOSTSVR1	DBDR
Control Tables - CRP	Control Tables - CRP	HOSTSVR1	DBDR
Control Tables - CRP - DNT	Control Tables - CRP - DNT	HOSTSVR1	DBDR
Control Tables - PROD	Control Tables - PROD	HOSTSVR1	DBDR
Control Tables - PROD - DNT	Control Tables - PROD - DNT	HOSTSVR1	DBDR
Control Tables Replicated	Control Tables Replicated	HOSTSVR2	JDBNET
Data Dictionary - B7332	Data Dictionary - B7332	HOSTSVR1	DBDR
HOSTSVR1	HOSTSVR1 - B7332 Server Map	HOSTSVR1	DBDR
HOSTSVR1 - Logic	HOSTSVR1 - B7332 Server Map	HOSTSVR1	DBDR
HOSTSVR2	HOSTSVR2 - B7332 Server Map	HOSTSVR2	JDBNET
HOSTSVR1 - Logic	HOSTSVR2 - B7332 Server Map	HOSTSVR2	JDBNET
Object Librarian - B7332	Object Librarian - B7332	HOSTSVR1	DBDR
System - B7332	System - B7332	HOSTSVR1	DBDR
System - B7332 - DNT	System - B7332 - DNT	HOSTSVR1	DBDR
Versions - CRPB733 - DNT	Versions - CRPB733 - DNT	HOSTSVR1	DBDR
Versions - PRODB733 - DNT	Versions - CRPB733 - DNT	HOSTSVR1	DBDR

Note

It is important that these changes are not made to the System - B7332 Data Source table.

20.5.5 Modifying the AS/400 server jde.ini file

If the defined environments in the Server Object Configuration Master (OCM) table F986101 differ from one host to the next, you need to add the following statements to the AS/400 server jde.ini file:

```
[SERVER ENVIRONMENT MAP]      <- Add these statements, where the
DEV733=PRD733                  remote_environment equals the local_environment
CRP733=PRD733
```

The following example describes a scenario requiring these statements for two AS/400 servers: HOSTSVR1 and HOSTSVR2:

- The HOSTSVR1 server contains the System - B7332 Data Source library shared by all OneWorld systems.
- The HOSTSVR1 server only consists of a PRODB733 path code and PRD733 environment. The server OCM for this machine contains only a PRD733 environment.

- The HOSTSVR2 server consists of a DEVB733 and CRPB733 path code and their corresponding environments. The server OCM for this machine contains the following environments CRP733, DEV733, and TST733.

These jde.ini Server Environment Map statements are added to the HOSTSVR1 AS/400 server jde.ini file to allow HOSTSVR2 to establish a connection using the JDBNet kernel process. When the initial JDBNet request is received by HOSTSVR1, it reads, among other things, its own server OCM to define the OneWorld kernel process characteristics for this connection.

Thereafter, the tables are accessible through this dedicated kernel process as defined by the Data Source Master table. For example, the System - B7332 Data Source resides on server HOSTSVR1 in library SYSB7332, and a JDBNET DLL driver is used to access it.

20.5.6 JDBNet dependencies

Since only one OneWorld host server consists of your System - B7332 Data Source, we call this system the *anchor server*. Secondary OneWorld servers depend on the anchor server for shared data. If the secondary server is an Intel Windows NT machine (also Alpha NT to MS SQL), it relies on third-party database drivers to establish a connection to a OneWorld supported RDBMS. However, all non-Intel Windows NT machines require a JDBNet process to establish the same connection.

Due to this dependency, you must start OneWorld services on the anchor server prior to starting these services on secondary servers. The secondary servers always depend on the anchor server. Therefore, if OneWorld services are stopped and restarted on the anchor machine, the same should be done on the secondary server.

Chapter 21. Performance management

The key to good system performance is to provide sufficient system resources in a balanced configuration. Here, the available processor capacity, memory capacity, disk I/O capacity, and network communications capacity are individually sufficient to handle the application workload, and in balance with one another, so that no performance bottlenecks exist. Performance management, the technique used to make sure this is the case, is an ongoing, cyclical process. It consists of four key steps:

1. Measure
2. Analyze
3. Adjust
4. Re-measure

For good performance of your system, these steps should be repeated on a regular basis. The results should be used to predict and schedule required system upgrades as well as to resolve specific performance problems.

OneWorld is an ODBC-based client/server ERP application that extensively uses SQL on the AS/400 system. This chapter introduces you to a performance management methodology for optimally running the OneWorld application on the AS/400 system. The approach we recommend is based on the ongoing measurement and analysis of system performance to enable you to perform capacity planning and specific performance problem analysis. Performance tuning concepts, tuning techniques, and tuning recommendations are described, along with AS/400, OneWorld, and third-party performance analysis tools that can be used to measure and analyze various components of system and application performance.

21.1 Performance concepts

This section presents an overview of some key factors that affect system, application, and network performance.

21.1.1 Queuing concepts

The work of a single job, or the transactions within that job, is comprised of several tasks. The invitation to perform the work required by a task is called a *request*. The requested work is performed by a server. The time taken to complete the requested work of the task is called *service time*.

Queuing is a concept that applies equally to requests waiting for computer resources and to people waiting in line at the supermarket or bank. In general, the length of time it takes to get a work request completed, whether it be a request to complete the purchase at the supermarket counter, complete a transaction at the bank, perform a disk I/O operation, or use the CPU, depends on three primary parameters:

- The number of “waiters” in the line ahead of a new request
- The number of servers responding to requests
- The service time to complete a request given to the server, which is a function of the speed of the server and the amount of work to do

Consider a service point where certain tasks are performed for requestors of service. Generally, requests for service are responded to in the order in which they arrive. Therefore, those arriving first are responded to first and leave the service point first.

If the rate of arrival of requests is greater than the rate at which they leave after being serviced, a queue is built at the server. The total response time to have a request serviced is the sum of:

- The time spent in the queue waiting to be serviced
- The time it takes the server to perform the requested work

When the queue grows longer, more time is spent waiting in the queue, and the total time taken for a request becomes longer.

The following basic principles govern queuing:

- A single server can service only one request at a time.
- Multiple concurrent requests are queued for service.
- The higher the server utilization is, the longer the queue is, and the longer queue waiting time and total response time are.

In the AS/400 environment, examples of service requestors are:

- Applications
- System tasks

Examples of service providers are:

- CPU
- I/O processors
- Disk arms

The equivalent functions of requestors and servers are also present within the client system and within the communications network.

When requests arrive randomly at a single server with a single queue, there is an equation that predicts the expected service times as a function of server utilization with a good degree of accuracy over time. This equation expresses the expected service times as a multiple of the time it takes to process a request once it has finished waiting in the queue and is actually processed by the server. The equation is:

$QM = 1 / (1 - U)$, where U = utilization and QM is the Queuing Multiplier

For example, if server utilization is at 50%, $1 / (1 - .5) = 2$, this indicates that a new request arriving in the queue is expected to take twice as long to be completed as it would if the server were not being utilized at all.

Response time is directly related to queue length, and the queuing multiplier expresses the effect of queuing on response times. A graph of the queuing multiplier against utilization of the server is shown in Figure 271.

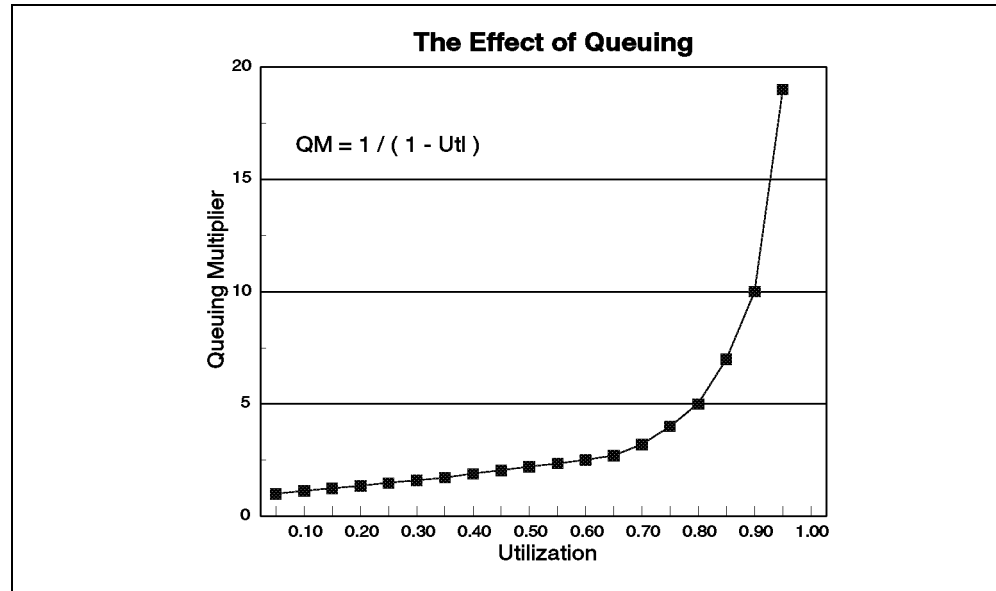


Figure 271. Queuing multiplier = 1 / 1 - U

As the utilization of a server increases (more work for the server), queuing can account for a much longer elapsed time for work (or request) completion.

The queuing multiplier is an important factor when projecting the impact of adding work or additional hardware on current system performance. Systems with performance problems often show resources with high queuing multiplier factors.

The simplified queuing theory discussed here assumes a single queue of requestors and a single server. In the high-end models of the AS/400 product range, multiprocessor (N-way) systems have more than one central processor executing instructions, even though there is only a single queue of requestors (Task Dispatching Queue). In this situation, the increased number of servers reduces the queuing multiplier and the average queue length somewhat, but the effect of queuing on response times is still a significant factor.

21.1.2 Response time curve

Response time is the elapsed time between the request for a service and the completion of that service. For a OneWorld user, the interactive response time is the time between sending a request to the server and receiving a response screen.

The queuing multiplier is a measure of the effect of queue length on response time, and U is the utilization of the resource providing the service.

Another important concept is highlighted by the graph in Figure 271. The curve shows the utilization at various rates and the significance of the knee of the curve. The knee of the curve is the point where a change in utilization produces a correspondingly greater change in the queuing multiplier. That is, the change along the Y-axis (queuing multiplier) is significantly greater than the change along the X-axis (utilization). The knee of this curve is the maximum utilization point to which a certain resource should be driven. After this knee, service time becomes less stable and may increase dramatically for small utilization increases.

Not all resources react the same. There are different recommended maximum values for the different resources, such as CPU, disk, memory, controller, remote line, IOPs, and so on.

AS/400 Performance Tools, SC41-5340, provides more queuing information. The graph shows a simplified queuing formula and a curve derived from it that highlights the effect of increasing utilization on the queuing multiplier for a single server.

21.1.3 Client/server response time

In a client/server environment, the response time perceived by the user is the total response time of the following service providers:

- **Client system:** When a user at a client system, such as a PC, requests information, that request is first processed by the PC and translated to a request to the server system.
- **Communication network** (to the server system): The request is sent through the line to the server (such as a database or application or file server).
- **Server system:** The server system accepts the request and performs the requested functions.
- **Communication network** (from the server system): The server response is sent back to the client.
- **Client system:** The client receives the information, performs further processing as necessary, and presents the final response to the user's request.

Therefore, the total response time experienced by a client/server application user is the sum total of the service times of the:

- Client
- Server
- Network

This is further complicated if the OneWorld environment involves multiple distributed servers. In this case, the perceived response time is the maximum of the response time from the individual servers if a single form is requesting information from the multiple servers. If a WAN is included in one of the communications segments, the response time is usually the predominant component. All efforts should be made to reduce the number of transmissions of data across the WAN to keep the perceived response time to a minimum.

Although this chapter is mainly intended to cover details on AS/400-related performance issues, tuning parameters of Windows NT-based clients and the communication network are also briefly covered.

Typically, a server system functions in an environment with multiple requestors. The response time experienced by a requestor is affected not only by the function of the particular task, but also by the workload introduced by other concurrent requestors and the relative servicing priority assigned to them.

Client PCs, on the other hand, are single-user systems where the contention for resources is minimal. However, with the introduction of multi-tasking operating systems and more concurrent activity on the PCs, resource contention is

becoming a significant contributor to overall client/server performance. The number of times information has to move between the client and server (communications flows) before a response is completed also increases the response time.

21.2 Tuning

This section describes the recommended tuning parameters for optimal OneWorld performance for the client, the network, and the AS/400 server.

21.2.1 Client tuning

This section discusses setting up of performance related parameters for OneWorld on a Windows NT client.

21.2.1.1 Client Access Express ODBC setup for performance

ODBC configuration can be done from the ODBC Administrator of your client PC. For details on ODBC Data Source setup, refer to “OneWorld and ODBC Data Sources” on page 191. To configure ODBC, complete the following process:

1. Set IP address Lookup Frequency to *After Startup*.

With this option, the first ODBC connection attempt after a PC is booted generates an IP address lookup. All subsequent connection attempts will assume that the IP address from the first address lookup is still correct. The benefit is that the subsequent connection attempts will gain performance.

2. Set Where to Lookup Remote Port to *Standard*.

3. Select **Enable Extended Dynamic Support** to enable SQL package support and provide the best performance.

Extended dynamic (package) support allows a prepared SQL statement to be stored on the AS/400 system. On subsequent uses of the same SQL statement, the database management system (DBMS) recognizes that the statement has been run before. The DBMS uses the information that was saved in the package to skip a significant part of the processing.

All statements that contain parameter markers (“?”) can be stored in a package. The following types of SQL statements are also stored in the package even when they do not contain parameter markers: positioned UPDATE and DELETE, INSERT with subselect, and DECLARE PROCEDURE.

4. Enable **lazy close support**. This will reduced system overhead and elapsed time.
5. Disable **Prefetch on Execute**.

This is a change from previous recommendations to prevent possible application problems. OneWorld is not designed to use the prefetch option.

6. Set Record blocking Type to **Block except if FOR UPDATE OF specified**. This will reduce overhead of the network, AS/400 IOP, and CPU.

7. Set Record blocking Size to **8 KB**, initially. This is data and application dependent, so some testing is required to determine what setting is optimal for a particular installation.

8. Enable **local package caching**.

9. Data compression is enabled for the ODBC Data Source for Central Objects. Testing is required to determine the appropriate setting for other data sources.
10. Set OS/400 library view to **Default library list**. *Do not* select All libraries on the system because it will severely impact performance.

21.2.1.2 jde.ini file setting on Windows NT Client

A file called jde.ini resides on an Windows NT client in the WINNT directory that is the major interface for the user to control the performance of OneWorld from the client perspective. This file is created on the Deployment Server and pushed out to the clients during the installation process.

Overall settings considerations

Refer to "Overall settings considerations" on page 585.

21.2.1.3 Debug and logging

A production client should not have the JDEDEBUG.LOG file turned on in the jde.ini or an SQL trace file turned on through the ODBC setup.

21.2.1.4 Lock manager

The lock manager function performs a useful function, but at the cost of increased traffic to the OneWorld server, which is running the lock manager kernel function. The cost of this functionality should be carefully weighed in a WAN environment. The lock manager function is individually controlled by a setting in each client's jde.ini file.

21.2.1.5 Security

Row Security=DEFAULT should be set in the [SECURITY] section to have row security be enforced by WHERE clauses. If its value is NO_DEFAULT, row security will not be done from the WHERE clause. Instead, the records are fetched first and then filtered. DEFAULT is more efficient and more reliable.

21.2.2 Network tuning

The network design, setup, and tuning is key to achieving good performance. The network frequently can be the slowest link in the system. This needs to be considered in the original design and implementation to achieve the performance goals. Typically this is achieved by reducing the number of turns across the WAN network to a minimum at the expense of potentially increasing network traffic on the LAN.

For TCP/IP communications, the key performance parameters that can be changed on the AS/400 system are the maximum transmission unit size and the send and receive buffer size. Recommendations for a typical LAN environment are:

- **Maximum Transmission Unit (MTU) Size:** Specifies the maximum size (in bytes) of IP datagrams that can be transmitted through this route.

Use the `CHGTCPRIE` command to set the MTU value to *IFC (means that the maximum transmission unit (MTU) is the MTU of the interface that is associated with this route). The display appears as shown in Figure 272.

```

Change TCP/IP Route (CHGTCPRTE)

Type choices, press Enter.

Route destination . . . . . > *DFTROUTE
Subnet mask . . . . . > *NONE
Type of service . . . . . *NORMAL      *MINDELAY, *MAXTHRPUT...
Next hop . . . . . > '172.16.1.1'
Preferred binding interface . . *NONE
Maximum transmission unit . . . > *IFC      576-16388, *SAME, *IFC
Route metric . . . . . *SAME      1-16, *SAME
Route redistribution . . . . . *SAME      *SAME, *YES, *NO
Duplicate route priority . . . . *SAME      1-10, *SAME

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 272. Change TCP/IP Route display

- **Buffer Size:** Use the CHGTCPA command to set the TCP/IP buffer size to 64K. The display appears as shown in Figure 273.

```

Change TCP/IP Attributes (CHGTCPA)

Type choices, press Enter.

TCP keep alive . . . . . 120      1-40320, *SAME, *DFT
TCP urgent pointer . . . . . *BSD      *SAME, *BSD, *RFC
TCP receive buffer size . . . . 65536    512-8388608, *SAME, *DFT
TCP send buffer size . . . . . 65536    512-8388608, *SAME, *DFT
UDP checksum . . . . . *YES      *SAME, *YES, *NO
IP datagram forwarding . . . . . *NO      *SAME, *YES, *NO
IP source routing . . . . . *YES      *SAME, *YES, *NO
IP reassembly time-out . . . . . 120      5-120, *SAME, *DFT
IP time to live . . . . . 64      1-255, *SAME, *DFT
ARP cache timeout . . . . . 5      1-1440, *SAME, *DFT
Log protocol errors . . . . . *NO      *SAME, *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 273. Change TCP/IP Attributes display

Ensure that the modes for Ethernet are matched correctly. For example, if a switch is set to full duplex, the AS/400 system is set to full duplex.

21.2.3 AS/400 server tuning

The AS/400 server plays a crucial role in the overall performance of OneWorld applications. Server performance can be impacted by:

- Hardware configuration and utilization of hardware components
- System tuning and workload management

- OneWorld application tuning
- Configuration of SQL package libraries

This section discusses performance requirements and recommendations for all of these areas.

21.2.3.1 AS/400 server hardware

Hardware plays a key role in performance. If your hardware is overused, you can expect degraded performance. Some of the hardware components that affect performance are:

- Server CPU:
 - Utilization
 - CPU speed
 - Number of processors
- Server disk:
 - Number of disk accesses
 - Disk performance
 - Disk storage capacity
 - Total number of disks
- Server memory:
 - Amount of paging
 - Cache
- Server Input/Output Processor (IOP):
 - Utilization
 - IOP capacity
 - Efficiency of frame size
 - Amount of error recovery necessary

The server CPU handles requests from a client. If the CPU utilization is exceptionally high, this causes queuing of any incoming requests until there are enough CPU cycles free to process the next request. This means that much of the response time is spent in waiting for the CPU.

Server main memory determines the amount of pages that can be processed. The more memory you have, the more pages can be brought into memory for processing, and the better the CPU is utilized. If you have a memory constraint, a fast CPU may do only a fraction of the workload of which it is capable.

Memory also plays an important role called *caching*. Caching is the process of bringing data into memory before it is needed, therefore, reducing the time required to process the data.

Hard disk access is crucial to response time. DASD performance is controlled by the disk subsystem, the actuator arms, and the capacity of the disks. The disk actuator arms should never exceed 40% busy (in real life, they are much less). The more actuator arms there are in the disk array, the better the performance is.

For high availability, DASD should be protected with RAID or mirror protection. Mirrored protection offers the best performance. However, it usually costs more than a RAID-5 configuration. RAID-5 is lower in cost, but can result in some performance penalty for heavy write intensive environments. In a RAID-5

environment, a single write request requires a read to another drive in the strip set and two writes. Therefore, RAID-5 makes drives slightly slower than unprotected drives. In the case of mirroring, the writes may be slightly slower. The reads are generally faster than the unprotected drives. For more details on RAID-5 and mirroring, refer to *Backup and Recovery*, SC41-5304.

For further details on AS/400 configuration and hardware sizing, see Chapter 8, “Pre-installation and planning” on page 139.

For backup, 3570 or 3590 tape drives are both excellent high-speed, high-capacity drives. Refer to *Backup and Recovery*, SC41-5304, for backup strategies.

Here are some recommended performance threshold guidelines for AS/400 resource utilization for OneWorld Software:

- **CPU Utilization up to 20 priority:** 65% to 70%
- **Disk IOP Utilization:** 20%
- **Disk CPU utilization:** 20%
- **Communications line utilization:** 30%
- **DASD arm utilization:** 40%

AS/400 server model behavior

Traditional (non-server) AS/400 models have a single Commercial Processing Workload (CPW; an AS/400 relative performance measurement) value that represents the maximum workload that can be applied to that model. This CPW value is applicable to either an interactive workload, a client/server workload, or a combination of the two (like in a OneWorld and WorldSoftware coexistence environment).

On a server or custom server model, there are two CPW values. The larger value represents the maximum workload the model could support if the workload were entirely client/server. The smaller CPW value represents the maximum workload the model can support if the workload is entirely interactive.

Server models were designed for a client/server workload and to accommodate a small interactive workload. When the interactive workload exceeds the maximum interactive CPW, the client/server processing performance of the system becomes impacted. Microcode task CFINTnn, for all AS/400 models, is an indication of the time to handle interrupts and task switching. For server models, it is also an indication of how much the system performance is impacted by the interactive processing. The amount of this impact depends on the AS/400 model and the release of OS/400. For example, the impact will be lower on new AS/400 Models 2XX and 8XX with OS/400 V4R5, than on previous models with previous OS/400 releases. For details, refer to *Performance Capabilities Reference VXYR* (where X is the version, and Y is the OS/400 release that you are interested in) at: <http://publib.boulder.ibm.com/pubs/html/as400/online/chgfrm.htm>

21.2.3.2 Basic AS/400 tuning commands

This section explains the basic level of system performance tuning you can do with the AS/400 system commands.

Work with System Values (WRKSYSVAL)

This section describes a few values that should be reviewed to optimize overall AS/400 system performance. Refer to *OS/400 Work Management Guide*,

SC41-5306, for more information on setting AS/400 system values. Here are some of the system values you may want to consider:

- **QACTJOB**: Initial number of active jobs
- **QADLACTJ**: Additional number of active jobs
- **QADLTOTJ**: Additional number of total jobs
- **QBASACTLVL**: Base storage pool activity level
- **QBASPOOL**: Base storage pool minimum size
- **QDYNPTYSCD**: Dynamic priority scheduler
- **QJOBMSGQFL**: Job message queue full action
- **QJOBMSGQMX**: Maximum size of job message queue
- **QMCHPOOL**: Machine storage pool size
- **QPFRAJ**: Performance adjustment
- **QTOTJOB**: Initial total number of jobs
- **QTSEPOOL**: Time slice end pool
- **QQRYPDEGREE**: Parallel processing degree

The scope of these system values is obviously over the entire system. For example, the possible values for QPFRAJ and the function of each value are:

- **0**: Switch off automatic tuning for all values, which means the users have to manually set the performance related values.
- **1**: Set up the system only at IPL time.
- **2**: Set up the system at IPL time and tune dynamically.
- **3**: Tune dynamically, but do not set up the system at IPL time.

Work with System Status (WRKSYSSTS)

This function provides information on AS/400 system resource utilization for a particular time interval, including:

- Disk capacity and using the system auxiliary storage pool (ASP1)
- Faulting rate per second in each memory pool
- Queuing jobs running in each memory pool

Select an elapsed time of at least five minutes for an interactive analysis of AS/400 resource utilization (F10 restarts elapsed time, and F5 extends the elapsed time). Run this command during typical levels of activity. Check the items that are highlighted in Figure 274.

```

Work with System Status
ASM23
07/14/98 18:06:03
% CPU used . . . . . :    50.9  Auxiliary storage:
Elapsed time . . . . . : 00:00:05  System ASP . . . . . : 26.35 G
Jobs in system . . . . . :    442  % system ASP used . . . : 94.3976
% addresses used:  Total . . . . . : 30.54 G
  Permanent . . . . . :    .009  Current unprotect used : 2381 M
  Temporary . . . . . :    .015  Maximum unprotect . . . : 2493 M

Type changes (if allowed), press Enter.

System  Pool  Reserved  Max  -----DB-----  ---Non-DB---
Pool  Size (K)  Size (K)  Active  Fault  Pages  Fault  Pages
  1    350000  159388  +++++  .0    .0    .0    .0
  2    3342048  0    187    .0    .0    .0    .0
  3     2256  0     1     .0    .0    .0    .0
  4    500000  0     80    .0    .0    1.2  1.3

Command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F10=Restart
F11=Display transition data  F12=Cancel  F24=More keys

Bottom

```

Figure 274. Work with System Status display (Part 1 of 2)

Note the following parameters:

- **% CPU Used:** This indicates the amount of CPU used during the elapsed time. If any batch processes were running at the time, this value can be high. If CPU usage is sustained at a high level, you may have a saturated processor.
- **% System ASP Used:** This indicates the amount of disk used of the total available in the system ASP. This value includes the amount of disk space occupied by the temporary objects created during normal AS/400 operations.
- **Total:** This refers to the total disk space in the entire system and excludes any disk space that was used in disk protection such as RAID-5 or mirroring.
- **Current Unprotected Used:** This indicates the amount of disk currently used by temporary objects.
- **Maximum Unprotected Used:** This indicates the maximum amount of disk used by temporary objects since the last IPL.
- **DB and Non-DB Faulting:** This is measured in faults per second and is the only measurement to determine how the AS/400 system uses the available memory in each pool. Make sure you review the total number of faults per second in each pool and not the number of pages.

Use F11 to view information on the AS/400 job transitions occurring in each memory pool.

```

Work with System Status
ASM23
07/14/98 18:06:03
% CPU used . . . . . : 50.9 Auxiliary storage:
Elapsed time . . . . . : 00:00:05 System ASP . . . . . : 26.35 G
Jobs in system . . . . . : 442 % system ASP used . . : 94.3976
% addresses used: Total . . . . . : 30.54 G
Permanent . . . . . : .009 Current unprotect used : 2381 M
Temporary . . . . . : .015 Maximum unprotect . . : 2493 M

Type changes (if allowed), press Enter.

System Pool Reserved Max Active-> Wait-> Active->
Pool Size (K) Size (K) Active Wait Inel Inel
1 350000 159388 +++++ .0 .0 .0
2 3342048 0 187 136.1 .0 .0
3 2256 0 1 .0 .0 .0
4 500000 0 80 31.4 .0 .0

Bottom

Command
====>
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F10=Restart
F11=Display pool data F12=Cancel F14=Work with subsystems F24=More keys

```

Figure 275. Work with System Status display (Part 2 of 2)

- **Wait->Inel:** This measures the transition of jobs from the wait state to the ineligible state where a job is ready to run but cannot secure an activity level. This indicates that the value set for *maximum active* jobs in the memory pool is too low.

Work with Disk Status (WRKDSKSTS)

The Work with Disk Status (WRKDSKSTS) command shows the AS/400 system's disk activity and helps to determine any potential impact on performance. The display is shown in Figure 276.

```

Work with Disk Status
ASM23
07/14/98 18:25:29
Elapsed time: 00:00:11

Unit Type Size % I/O Request Read Write Read Write %
(M) Used Rqs Size (K) Rqs Rqs (K) (K) Busy
1 6606 1967 94.3 .1 4.0 .0 .0 4.0 4.0 0
2 6606 1967 94.4 .3 4.0 .2 .0 4.0 4.0 0
3 6607 4194 94.3 .1 4.0 .0 .0 4.0 4.0 0
4 6607 4194 94.6 .1 4.0 .0 .0 4.0 4.0 0
5 6606 1475 94.3 .2 4.0 .1 .0 4.0 4.0 0
6 6606 1475 94.4 .4 4.0 .0 .3 4.0 4.0 0
7 6606 1967 94.3 .1 4.0 .0 .0 4.0 4.0 0
8 6606 1967 94.3 .0 4.0 .0 .0 4.0 .0 0
9 6606 1475 94.4 .5 4.0 .5 .0 4.0 .0 0
10 6606 1475 94.3 .2 4.0 .0 .1 4.0 4.0 0
11 6607 4194 94.3 .1 4.0 .0 .0 4.0 4.0 0

Bottom

Command
====>
F3=Exit F5=Refresh F12=Cancel F24=More keys

```

Figure 276. Work with Disk Status display

On this display, note the following parameters:

- **% Used:** This indicates the amount of space used on a disk unit listed. The amount should not exceed 90% on any disk except the first disk (or the first two disks if the internal disks are mirrored). Above this number, you may see the average disk arm seek time increase significantly due to disk space fragmentation.

Be sure you have adequate spare disk space to allow for growth in the database.

- **% Busy:** This is a key factor in disk performance and should remain under 40%.

Work with Active Job (WRKACTJOB)

This command displays all of the jobs running on the AS/400 system within each of the active subsystems and shows you detailed information including:

- The subsystem
- User identifier
- Job type
- Memory pool
- Job priority
- CPU usage (%)
- CPU usage (seconds)
- Disk I/O count

The displays that appears are shown in Figure 277 and Figure 278 and Figure 279 on page 574.

```

Work with Active Jobs
ASM23
07/15/98 10:21:03
CPU %: .3 Elapsed time: 00:02:00 Active jobs: 166

Type options, press Enter.
 2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
 8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
QBATCH QSYS SBS .0 DEQW
QCMN QSYS SBS .0 DEQW
QCTL QSYS SBS .0 DEQW
 QSYSSCD QPGMR BCH .0 PGM-QEZSCNEP EVTW
QINTER QSYS SBS .0 DEQW
 QPADEV0019 YESSONG INT .0 CMD-WRKACTJOB RUN
 QPADEV0020 SDICKEY INT .0 MNU-MAIN DSPW
 QPADEV0022 V2CDINR INT .0 * -CMDENT DSPW
QSERVER QSYS SBS .0 DEQW

Parameters or command
====>
F3=Exit F5=Refresh F10=Restart statistics F11=Display elapsed data
F12=Cancel F23=More options F24=More keys
More...

```

Figure 277. Work with Active Jobs display (Part 1 of 3)

Use F11 to show additional information on the jobs running on the AS/400 system.

```

                                Work with Active Jobs                                ASM23
                                                                07/15/98 10:21:03
CPU %:      .3      Elapsed time:  00:02:00      Active jobs:  166

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files  13=Disconnect ...

-----Elapsed-----
Opt Subsystem/Job  Type  Pool  Pty    CPU  Int  Rsp  AuxIO  CPU %
QBATCH          SBS    2    0     .1   0    0    0     .0
QCMN            SBS    2    0     .6   0    0    0     .0
QCTL           SBS    2    0     .7   0    0    0     .0
  QSYSSCD       BCH    2   10     .8   0    0    0     .0
QINTER          SBS    2    0     .9   0    0    0     .0
  QPADEV0019    INT    4   20     .3   5    .0    3     .0
  QPADEV0020    INT    4   20    2.4   0    .0    0     .0
  QPADEV0022    INT    4   20     .0   0    .0    0     .0
QSERVER         SBS    2    0    1.1   0    0    0     .0
                                                                More...

Parameters or command
====>
F3=Exit      F5=Refresh      F10=Restart statistics  F11=Display status
F12=Cancel   F23=More options  F24=More keys

```

Figure 278. Work with Active Jobs display (Part 2 of 3)

All OneWorld online jobs run in the QSERVER subsystem as QZDASOINIT jobs. If you are only interested in the status of OneWorld online jobs, you can limit the information to be shown by typing:

WRKACTJOB SBS(QSERVER)

```

                                Work with Active Jobs                                ASM23
                                                                07/15/98 10:33:49
CPU %:      .4      Elapsed time:  00:01:02      Active jobs:  170

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files  13=Disconnect ...

Opt Subsystem/Job  User      Type  CPU %  Function  Status
QSERVER          QSYS     SBS   .0     DEQW
  QPWFSERVSD     QUSER    BCH   .0     SELW
  QPWFSERVSO     QUSER    PJ    .0     DEQW
QSERVER          QPGMR    ASJ   .0     EVIW
  QZDASOINIT     QUSER    PJ    .0     DEQW
  QZDASRVSD      QUSER    BCH   .0     SELW

                                                                Bottom

Parameters or command
====>
F3=Exit      F5=Refresh  F10=Restart statistics  F11=Display elapsed data
F12=Cancel   F23=More options  F24=More keys

```

Figure 279. Work with Active Jobs display (Part 3 of 3)

21.2.3.3 AS/400 workload and memory management

Workload and memory management are important for good performance on AS/400 systems. This section provides key recommendations on workload and memory management for OneWorld environment.

Workload management

It is important to manage the jobs running on the system to minimize peak workload. Low priority batch jobs may be scheduled to run at a lower system priority. Avoid the unnecessary creation of job logs because they often cause increasing the workload on a system. Section 22.3.2, “Logs” on page 730, explains how you can change the log level of jobs to produce job logs. Make sure that for normal production environment, the log level is set to `LOG(4 00 *NOLIST)`. Also, when ending the subsystem QSERVER, always use `ENDSBS (*NOJOBLOG)`. This ensures that no job logs are created when ending the subsystem. This option can significantly reduce the amount of time necessary to complete the `ENDSBS` command. However, if a problem occurs in a job, there will be no job log to record the problem, which may make problem diagnosis difficult or impossible.

The general recommendations for workload management are:

- Run the Database Server programs at their default priority of 20.
- Prioritize the jobs to keep CPU utilization for the high priority workload (Database Server jobs, interactive workload) to less than 65 to 70%.
- In a OneWorld environment involving heavy UBE or batch processing, keep the utilization of the high priority workload to no more than 40%. This ensures that CPU cycles are available for the low priority batch jobs.
- Run batch jobs in their own, separate pool.
- If possible, schedule the low priority batch jobs to run during off peak periods.
- If using a multi-processor AS/400, use multiple processors by appropriately scheduling or multi-threading the batch jobs.
- Minimize job logs (print on failure only).

Memory

The default memory pool for the Database Server jobs (QZDASOINIT) is *BASE. Many of the AS/400 system tasks are also performed in the *BASE pool. This can result in memory contention for the system and server jobs.

Follow these recommendations for memory management:

- Change the QSERVER subsystem to run the prestarted QZDASOINIT jobs in a separate pool. For more details on this, see “Assigning a separate storage pool for QZDASOINIT jobs” on page 576.
- Adjust the values for the initial number of jobs, number of available jobs, and the additional number of jobs on the `CHGPJE` command.
- Change the system value `QPFRADJ` to 2 or 3 to help in initially setting up the pool size and activity level of the main storage pool running QZDASOINIT jobs. Once the pool size stabilizes, turn the autotuner off by changing the system value `QPFRADJ` to 1. Monitor the pools for excessive faults and make appropriate adjustments based on the steps described in “User pool faulting guidelines” on page 578.

- Turn on Expert Cache for the UBE and batch pools. See “Turning ‘ON’ Expert Cache” on page 578.
- Adjust the size of the UBE and QZDASOINIT pools based on the day and night time workloads. For example, at night, you may want to give more memory to the UBE pool. On the other hand, during the day, you may want to allocate more memory to the QZDASOINIT pool. You can automate day and nighttime pool settings by running a CL program containing CHGSHRPOOL commands.

Assigning a separate storage pool for QZDASOINIT jobs

The Database Server job QZDASOINIT runs in the QSERVER subsystem and uses *BASE memory pool by default. It also uses job class QSYS/QPWFSEVER, which has the following values when shipped:

- Run priority = 20
- Time slice = 3000 msec

Depending on the number of server jobs running concurrently, perform the following tasks:

1. Create an additional memory pool in QSERVER subsystem with the appropriate activity level (roughly equal to the expected number of concurrent users of OneWorld application) using the CHGSBSD command (Figure 280).

```

Change Subsystem Description (CHGSBSD)

Type choices, press Enter.

Subsystem description . . . . . > QSERVER      Name
Library . . . . . *LIBL                      Name, *LIBL, *CURLIB
Storage pools:
Pool identifier . . . . . 2                    1-10, *SAME
Storage size . . . . . 9000                   Number, *BASE, *NOSTG.
Activity level . . . . . 6                    Number
+ for more values
Maximum jobs . . . . . *SAME                  0-1000, *SAME, *NOMAX
Text 'description' . . . . . *SAME

```

Figure 280. Change Subsystem Description

2. Change the prestart job entry using the CHGPJE command to select the additional pool (Figure 281).

```

Change Prestart Job Entry (CHGPJE)

Type choices, press Enter.

Subsystem description . . . . . > QSERVER      Name
Library . . . . . > QSYS          Name, *LIBL, *CURLIB
Program . . . . . > QZDASOINIT   Name
Library . . . . . *LIBL         Name, *LIBL, *CURLIB
User profile . . . . . *SAME     Name, *SAME
Start jobs . . . . . *SAME     *SAME, *YES, *NO
Initial number of jobs . . . . . *SAME   1-1000, *SAME
Threshold . . . . . *SAME     1-1000, *SAME
Additional number of jobs . . . . *SAME   0-999, *SAME
Maximum number of jobs . . . . . *SAME   1-1000, *SAME, *NOMAX

|
Maximum number of uses . . . . . *SAME   1-1000, *SAME, *NOMAX
Wait for job . . . . . *SAME     *SAME, *YES, *NO
Pool identifier . . . . . > 2      1-10, *SAME
Class:
Class . . . . . > QPWFSERVER   Name, *SAME, *SBSD
Library . . . . . > QSYS       Name, *LIBL, *CURLIB

```

Figure 281. Change Prestart Job Entry

Tuning the memory pools

Due to the large range of AS/400 processors and an ever increasing variance in the complexity of the user applications, paging guidelines for user pools are no longer published. Only machine pool guidelines and system wide guidelines (sum of faults in all the pools) are published. *OS/400 Work Management Guide*, SC41-5306, contains page fault guidelines for the machine pool and the total for all other storage pools defined on the system. These guidelines are based on system processor speeds and are listed based on model feature numbers. Refer to this manual for these guideline values. For all other user pools, each customer needs to track response time, throughput, and CPU utilization against the paging rates to determine a reasonable paging rate. Generally the faster the resource is, the higher is the fault threshold tolerance. The slower the resource is, the lower the threshold tolerance is.

There are two choices for tuning the user pools:

- Set system value QPFRADJ = 2 or 3. This automatically tunes the page faulting and activity levels of the Shared Pools through an internal AS/400 algorithm. This algorithm quickly determines which pool has a paging problem and reacts to the needs of those pools with problems. When the system value is set to either 2 or 3, usually the customers do not have to worry about pool tuning at all. They still may need to analyze whether they need to increase total main storage.
- Manually tune. Move storage around until the response times and throughputs are acceptable.

Change the system value QPFRADJ to 2 or 3 to help initially set up the pool size and activity level of the main storage pool running QZDASOINIT jobs. Once the pool size stabilizes, turn off the autotuner by changing the system value

QPFRAJ to 1. Monitor the pools for excessive faults and make appropriate adjustments. See “User pool faulting guidelines” for further information.

Turning ‘ON’ Expert Cache

When you switch on Expert Cache, the system automatically adjusts storage pool paging and determines the best approach for handling data in the pool. If objects are referred to sequentially, the system brings larger blocks of data into the memory and delays writing changes of data to the disk. This reduces the number of I/O operations and reduces the disk arm contention.

If objects are referred to randomly, the system does not bring in large blocks of data because that does not reduce the number of I/O operations.

We recommend that you turn on Expert Cache for the QZDASOINIT and the UBE job pools. Expert Cache can be used only for the shared pools.

To switch Expert Cache ON for a pool, use the AS/400 `WRKSHRPOOL` command, and change the paging option to `*CALC`.

```

                                Work with Shared Pools
                                System:  ASM23
Main storage size (K) . . :      3145728

Type changes (if allowed), press Enter.

      Defined   Max   Allocated   Pool   -Paging Option--
Pool   Size (K) Active Size (K)   ID   Defined Current
*MACHINE  300164  +++++  300164   1   *FIXED  *FIXED
*BASE     2499540   84    2499540   2   *FIXED  *FIXED
*SHRPOOL1 314568   137    314568   4   *CALC   *CALC
*SPOOL    31456     7     31456   3   *FIXED  *FIXED

                                                                More...

Command
===>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display tuning data
F12=Cancel

```

Figure 282. Work with Shared Pools screen

User pool faulting guidelines

Due to a large range of AS/400 processors and an ever increasing variance in the complexity of user applications, paging guidelines for user pools are no longer published. Only machine pool guidelines and system wide guidelines (sum of faults in all the pools) are published. Each customer needs to track response time, throughput, and CPU utilization against the paging rates to determine a reasonable paging rate.

To determine a reasonable level of page faulting in user pools, determine how much the paging is affecting the interactive response time or batch throughput. These calculations show the percentage of time spent doing page faults.

The following guidelines assume that interactive jobs are running in their own pool, and batch jobs are running in their own pool. All data can be gathered with the `STRPFRMON` command and printed with the `PRTSYSRPT` command.

- **Interactive:** Note the following fields:
 - **Flts:** Sum of the database and non-database faults per second during a meaningful sample interval for the interactive pool.
 - **Rt:** Interactive response time for that interval.
 - **DiskRt:** Average disk response time for that interval.
 - **Tp:** Interactive throughput for that interval in transactions per second (transactions per hour/3600 seconds per hour).

$FltRtTran = DiskRt * Flts / Tp$ (Average page faulting time per transaction)

$Flt\% = FltRtTran / Rt * 100$ (Percentage of response time due to fault)

If Flt% is less than 10% of the total response time, there is not much potential benefit of adding storage to this interactive pool. However, if Flt% is 25% or more of the total response time, adding storage to the interactive pool may be beneficial.

- **Batch:** Note the following fields:
 - **Flts:** Sum of database and non-database faults per second during a meaningful sample interval for the batch pool.
 - **BatchCPU%:** Batch CPU utilization for the sample interval. If higher priority jobs (other than the batch jobs in the pool you are analyzing) consume a high percentage of the processor time, Flt% is always low. This means adding storage will not help much, but only because most of the batch time is spent waiting for the processor. To eliminate this factor, divide Flt% by the sum of Flt% and BatchCPU% as shown in the following example:

$Flt\% = Flts * DiskRt * 100$

Percentage of time spent page faulting in the batch pool. If multiple batch jobs are running concurrently, you need to divide Flt% by the number of concurrently running batch jobs.

$NewFlt\% = Flt\% / (Flt\% + BatchCPU\%)$

This is the percentage of time the job is spent page faulting compared to the time it spends at the processor.

Again, the potential gain of adding storage to the pool needs to be evaluated. If Flt% is less than 10%, the potential gain is low. If Flt% is greater than 25%, the potential gain is high enough to warrant moving main storage into this batch pool.

Note

It is difficult to predict the improvement of adding storage to a pool, even if the potential gain calculated above is high. There may be a time where adding storage may not improve anything because of the application design. For these circumstances, changes to the application design may be necessary. Also, these calculations are of limited value for pools that have Expert Cache turned on. Expert Cache can reduce I/Os given more main storage, but those I/Os may be page faults.

This is expressed as a percentage of total space available in the system auxiliary storage pool (ASP1). When OneWorld and any other applications are active, the usage indicated by the displays include all permanent objects, as well as any

temporary objects created by the AS/400 system to manage system activity. Each prestarted job uses temporary disk space. Therefore, during a peak period, due to the large number of active prestarted jobs on the AS/400 system, the disk space usage may be much higher than in a non-peak period.

Review your disk usage during peak activity. Excessive disk space usage can impact performance, but it is not directly related to the percentage used, only the available disk space. Ensure that you allow adequate disk space for growth of the database and the usage does not exceed 80% during the peak activity period. Reduce the disk usage by:

- Periodically reorganizing the physical files that contain deleted records
- Saving and deleting “less frequently used” objects

21.2.3.4 OneWorld application tuning

OneWorld application tuning involves many disciplines. The application needs to be properly configured to accomplish all that the business needs dictate. However, it does not need to be configured to perform extra work because we may use this function someday. The work needs to be performed in as efficient method as possible. Sometimes this means that the work should be done locally, sometime on a distributed CPU. And, sometimes this means that the work should be performed at the central site on one or more CPUs.

On the AS/400 Database Server, the database needs to be tuned based upon the SQL statements actually implemented. This involves attempting to perform the database optimization only once and storing the results in an SQL package that is used for each query. The AS/400 Database Server also needs to be prepared to run the proper number of jobs to handle the ODBC requests. Every extra job on the AS/400 system has a big performance impact when it starts and when it terminates. However, the job concept and structure on the AS/400 system is what achieves the strengths and reliability of the AS/400 system. The Application Server must be tuned to run the proper number of threads.

Determining the proper number of ODBC connections

The OneWorld installation defaults create many individual ODBC connections. Each one of these connections becomes a separate job on the AS/400 system. Each connection also tends to point to a separate library for the data. The separate connections slightly help the SQL optimizer since it allows a separate SQL package to exist in each library. However, the separate connections cause more jobs to run in the system. Since only a few tables have the same name in the different libraries, it is possible to combine some of the libraries and, therefore, reduce the number of ODBC connections. This should not significantly reduce the execution times. However, it should reduce memory requirements, disk requirements, and start up and shutdown times.

Prestart job QZDASOINIT settings

Prestart jobs ensure that the Database Server jobs are running in QSERVER subsystem, and not in QCMN. When the server runs in QSERVER subsystem, it uses more efficient interfaces to send and receive data. OneWorld user applications, using the standard defaults, typically use six to eight ODBC connections. Each ODBC connection is connected to one QZDASOINIT program. The programs QZSCSRVS and QZSOSIGN are used for making the connection. Therefore, the initial value for QZDASOINIT can be set to five times the number of expected *concurrent* connected users. Keep in mind that the initiation and

termination of jobs are resource intensive activities. Therefore, a large number of these activities at a given time may impact system performance. At the same time, starting and stopping a subsystem (and IPL of the system) having many active jobs can be very slow.

To see the current settings, use option 10 from the `DSPSBSD QSERVER` command. The display shown in Figure 283 appears. Then, use option 10, followed by option 5, for the `QIWS/QZDASOINIT` program.

```

                                Display Prestart Job Entry Detail
                                System:  ASM23
Subsystem description:  QSERVER          Status:  ACTIVE

Program . . . . . :  QZDASOINIT
  Library . . . . . :  QIWS
User profile . . . . . :  QUSER
Job . . . . . :  QZDASOINIT
Job description . . . . . :  *USRPRF
  Library . . . . . :
Start jobs . . . . . :  *NO
Initial number of jobs . . . . . :  560
Threshold . . . . . :  25
Additional number of jobs . . . . . :  30
Maximum number of jobs . . . . . :  *NOMAX
Maximum number of uses . . . . . :  200
Wait for job . . . . . :  *YES
Pool identifier . . . . . :  1

More...

Press Enter to continue.

F3=Exit  F12=Cancel  F14=Display previous entry

```

Figure 283. Display Prestart Job Entry Detail

Use the following command to determine the number of current, average, and peak number of prestart jobs:

```
DSPACTPJ QSERVER QIWS/QZDASOINIT
```

The display shown in Figure 284 on page 582 appears.

```

                                Display Active Prestart Jobs
                                ASM23
                                02/24/99 10:59:08
Subsystem . . . . . : QSERVER      Reset date . . . . . : 02/22/99
Program . . . . . : QZDASOINIT   Reset time . . . . . : 09:06:39
Library . . . . . : QIWS         Elapsed time . . . : 08:00:00

Prestart jobs:
Current number . . . . . : 1,250
Average number . . . . . : 950
Peak number . . . . . : 1,540

Prestart jobs in use:
Current number . . . . . : 1,103
Average number . . . . . : 745
Peak number . . . . . : 1,540

More...

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F13=Reset statistics

```

Figure 284. Display Active Prestart Jobs example

Optimum setting for the initial number of jobs (INLJOBS), the threshold (THRESHOLD), the additional number of jobs (ADLJOBS), and the maximum number of uses (MAXUSE) are, therefore, very important for reasonable performance of the server. The recommended values are:

- Initial number of jobs: Average number of prestart jobs during the day

This number can be based on one full day's statistics reported on the DSPACTPJ display (refer to Figure 284). The last reset date, time, and the elapse time are displayed on the top right-hand corner of this display.

- Maximum Number of jobs: *NOMAX
- Additional Number of jobs: 30
- Threshold: 20
- Maximum number of uses: 200

Setting the initial number of jobs is a trade off. If you set it too high, the start and stop of the QSERVER subsystem may take too long. On the other hand, if you set it too low, the system may have to initiate too many new jobs, which may hurt the run time performance.

The above mentioned numbers are provided for initial prestart job entry setup. Depending on the number of users and the size of your AS/400 server, you may want to set these values differently.

The command to change the number of prestart jobs and other related values is CHGPJE.

```

Change Prestart Job Entry (CHGPJE)

Type choices, press Enter.

Subsystem description . . . . . > QSERVER      Name
  Library . . . . . > QSYS                Name, *LIBL, *CURLIB
Program . . . . . > QZDASOINIT          Name
  Library . . . . . *LIBL                Name, *LIBL, *CURLIB
User profile . . . . . *SAME             Name, *SAME
Start jobs . . . . . *SAME              *SAME, *YES, *NO
Initial number of jobs . . . . . *SAME   1-1000, *SAME
Threshold . . . . . *SAME               1-1000, *SAME
Additional number of jobs . . . . *SAME  0-999, *SAME
Maximum number of jobs . . . . . *SAME   1-1000, *SAME, *NOMAX

|
|
|
Maximum number of uses . . . . . *SAME   1-1000, *SAME, *NOMAX
Wait for job . . . . . *SAME            *SAME, *YES, *NO
Pool identifier . . . . . > 2           1-10, *SAME
Class:
  Class . . . . . > QPWFSEVER           Name, *SAME, *SBSD
  Library . . . . . > QSYS              Name, *LIBL, *CURLIB

```

Figure 285. Change Prestart Job Entry

Note

A background AS/400 job periodically monitors the Prestart activities. If at any time it finds the active prestart jobs to be much higher than needed (currently in use), it terminates a few unused prestarted jobs.

For OS/400 V4R3 and later, you do not need to stop and restart the subsystem or the prestart jobs when changing the prestart job values, because the system will automatically use the new values. If you are on an earlier release, you must restart the subsystem or end all of the prestarted jobs and restart them by using ENDPJ and STRPJ commands respectively.

You may have to increase the Max Active Job value for the pool in which the ODBC jobs are run. An indication of a need for this if there is an increase in the number of jobs in the INELIGIBLE status. The Max Active Jobs can be displayed and changed by using the WRKSYSSTS or WRKSHRPOOL commands. For an example of the WRKSYSSTS display, see “Work with System Status (WRKSYSSTS)” on page 570.

If the client PCs are *not* configured to bypass the signon check, the prestart jobs QZSOSIGN and QZSCSRVS initial values should also be adjusted similar to the way the job QZDASOINIT was adjusted. These two jobs run in the subsystem QSYSWRK. Their status can be displayed by entering the following commands:

```

DSPACTPJ QSYSWRK QIWS/QZSOSIGN
DSPQSYSWRK QIWS/QZSCSRVS

```

The following CL program can be used to periodically take a snapshot of the system activity to determine appropriate values. Each snapshot creates a set of spooled files:

```

PGM
LOOP:
DSPACTPJ SBS(QSERVER) PGM(QIWS/QZDASOINIT) + OUTPUT(*PRINT)
DLYJOB DLY(300)
GO TO LOOP
ENDPGM

```

Note that even if more jobs are needed than are configured, they are created as necessary. However, as previously mentioned, the starting and stopping of the extra jobs may cause extra system overhead.

Prior to V4R4, the maximum value allowed is 1000 using the screen interface. If a larger number is needed, you have to run a CL program that uses an API to set the initial value. A sample CL program to set the prestart jobs to greater than 1000 is shown below. This program is not required starting in V4R4, because the standard `CHGPJE` command now allows up to a maximum of 9999. This is the same value that the API supports.

Note

A PTF that supports the API to change subsystem description information is available that can be used to set the initial value to a number larger than 1,000:

- For V4R1, the PTF number is SF48069.
- For V4R2, the PTF number is SF48071.
- For V4R3 and later, this API is in the base operating system.
- For V4R4, this API is still available, but the same function can be done via the `CHGPJE` command.

Sample command and CL program

The following CL program and CMD can be used to provide a way of setting values larger than 1,000 for Releases V4R1 through V4R3.

To create this program and command, complete these steps:

1. Enter the CL source and CMD in an appropriate library member, for example:

```
JDEUTILS/QCLSRC, JDEUTILS/QCMDSRC, member CHGPJE1
```

2. Create the CL program with the `CRTCLPGM` command:

```
CRTCLPGM PGM(JDEUTILS/CHGPJE1) SRCFILE(JDEUTILS/QCLSRC)
```

3. Create the CMD with the `CRTCMD` command:

```
CRTCMD CMD(JDEUTILS/CHGPJE1) PGM(JDEUTILS/CHGPJE1)
SRCFILE(JDEUTILS/QCMDSRC)
```

4. Invoke the command by entering `JDEUTILS/CHGPJE1`, and then pressing PF4. Fill in a new value, and press Enter.

5. To verify change, use the `WRKSBSD` command:

```
WRKSBSD QSERVER
```

Select option **5**. Then, select option **10**. Then, select option **5** for the `QZDASOINIT` program.

The CL source code is shown here:

```
PGM PARM(&INITIALJOB)
/* SET NUMBER OF INITIAL JOBS FOR QZDASOINIT, THIS CAN BE USED WHEN VALUES */
/* GREATER THEN 1,000 ARE NEEDED */
DCL VAR(&INITIALJOB) TYPE(*CHAR) LEN(8)
DCL VAR(&SBSDDNAME) TYPE(*CHAR) LEN(20) +
VALUE('QSERVER QSYS')
DCL VAR(&FORMAT) TYPE(*CHAR) LEN(8) VALUE('SBSE0500')
DCL VAR(&PGMNAME) TYPE(*CHAR) LEN(20) +
VALUE('QZDASOINITQIWS')
DCL VAR(&CHGINFO) TYPE(*CHAR) LEN(150)
DCL VAR(&ERRCOD) TYPE(*CHAR) LEN(8) +
VALUE(X'0000000000000000')
DCL VAR(&RCO) TYPE(*CHAR) LEN(8) +
VALUE(X'0000000000000000')
DCL VAR(&BINNUM) TYPE(*CHAR) LEN(4)
CHGVAR VAR(%BIN(&BINNUM)) VALUE(&INITIALJOB)
CHGVAR VAR(%SST(&CHGINFO 1 4)) VALUE(X'00000001')
CHGVAR VAR(%SST(&CHGINFO 5 4)) VALUE(X'00000010')
CHGVAR VAR(%SST(&CHGINFO 9 4)) VALUE(X'00000003')
CHGVAR VAR(%SST(&CHGINFO 13 4)) VALUE(X'00000004')
CHGVAR VAR(%SST(&CHGINFO 17 4)) VALUE(&BINNUM) +
/* NUMBER OF INITIAL JOBS */
CALL PGM(QWDCSBSE) PARM(&SBSDDNAME &FORMAT +
&PGMNAME &CHGINFO &ERRCOD)
ENDPGM
```

The CMD source code is shown here:

```
CMD PROMPT('CHANGE QZDASOINIT INITIAL JOBS')
PARM KWD(INITJOBS) TYPE(*CHAR) LEN(8) MIN(1) +
ALWUNPRT(*NO) PROMPT('Initial QZDASOINIT jobs')
```

jde.ini server file settings

The INI file resides on an AS/400 system in the library referenced by the system data source (for OneWorld Release B73.3, the library is B733SYS). The file name is INI, and it contains the member JDE. This source file is 240 characters in length including the 12 reserved characters for maintenance. The settings in the INI file are the primary changes that the user can make that affect the performance of OneWorld.

To change the INI settings, you may either use the OneWorld SAW command or the AS/400 SEU command. A complete explanation of all of the INI settings can be found in Appendix A of the *OneWorld Server and Workstation Administration*.

Overall settings considerations

Use caution when changing the settings in the INI file. The keywords are case sensitive. Not all of the keywords are automatically included. If a keyword is not found, defaults are assumed.

Since it is easy to make a mistake, extra caution is necessary. It is good practice to first go through the entire file and change the setting for all server functions to be the name of a valid server. Simply perform a global change of “servername” to your primary OneWorld Application Server or Database Server name. Change the name of all library settings to be a valid and logical library name. The name should not be the proper one in the long run. However, this makes the error message stand out far more so than if the names are left in either the default or blank status.

A line in the INI file may be commented out by entering a semicolon (;) in the first position. It is good practice to keep the alternate choices of the few lines that are frequently changed.

Debug and logging

Make sure that the debug logs are turned off in the INI file for production usage. To accomplish this, set `Output=NONE` in the [DEBUG] section.

To ensure that you are looking at the message, place the print file that is opened, written to, and then closed for each line, into the debug log. If this method is not used, the current block of records may not be available for viewing. This is a good reason to double check that the debug logs are turned off in the INI file for production usage since logging can seriously impact performance on the AS/400 system. This impacts UBEs in particular.

Naming

It is good practice to make sure that the information in a commented out line is still logically correct. On rare occasions, the information is checked first before the comment status is read. This can be important if the commented out line was the name of a server, which needed to be validated on the network. In general, always replace “servername” with a valid server name on the network.

Kernels

Because a server can have multiple net processes (JDENet) and multiple dedicated internal processes, you must establish a configuration that provides an optimal number of processes on a server. This system administration concept is called *load balancing*. Load balancing allows OneWorld to maximize its overall performance by using the available processing power of the platform on which it is running.

The load balancing configuration is controlled by various interrelated runtime settings in the `jde.ini` file on the server. These settings are in the [JDENET] and applicable [JDENET_KERNEL_DEFx] sections. You can control two types of processes:

- Network jobs (JDENET_n)
- Dedicated kernel process jobs (JDENET_k)

You can set the number of network jobs to control how many users are simultaneously connected to each net job. For example, if you have 200 users and you want a net process for each 100 users, you can specify two net processes. The controlling setting is:

```
[JDENET]
MaxNetProcesses= variable
```

You can set the number of dedicated process jobs to control how many instances of each dedicated process type are running. For example, if you think that increasing the number of Type 2 dedicated processes may improve the processing performance for batch process (UBE) requests, you can increase the value. The controlling setting is contained in the definition for each dedicated process type (the x variable value in `JDE_KERNEL_DEFx`, where allowable values are from 1 to 12):

```
[JDENET_KERNEL_DEF x]
MaxNumberOfProcesses= variable
```

If you increase the number of individual dedicated processes, you must also increase the value of `MaxKernelProcesses` in the [JDENET] section. For example, consider the scenario where `maxKernelProcesses=12`:

```
[JDENET]
maxKernelProcesses=12
```

You increase the number of Type 2 processes from 1 to 2:

```
[JDENET_KERNEL_DEF2]
maxNumberOfProcesses=2
```

Then, in this case, you must increase the maxKernelProcesses value from 12 to 13.

The “call_object” kernel is typically used to scale the Application Server since this kernel controls the number of MBFs. It has been found that average CPU utilization, by the kernels, varies widely between a LAN and a WAN environment.

There are performance advantages to adding multiple kernels, up to a point. After such a point, overall performance is degraded.

An initial guideline for this kernel is:

- For a LAN-based environment, the optimal ratio of clients to Enterprise Server kernels is about 10:1, or 10 clients for each kernel process.
- For a WAN-based environment, the optimal ratio of clients to Enterprise Server kernels is about 20:1, or 20 clients for each kernel process.

The guideline for the UBE kernel is to set the number of processes one less than the number of CPUs. This also corresponds to the number of threads in the subsystem.

Note that the replication kernel supports only one process. If you have more than one, the changes go in a round-robin fashion. This means two changes to the same record may be replicated in the wrong order (second change, then first; instead of first, then second).

Security

Row Security=DEFAULT should be set in the [SECURITY] to have row security be enforced by WHERE clauses. If its value is NO_DEFAULT, row security will not be done from the WHERE clause. Instead, the records are fetched first and then filtered by the application. DEFAULT is more efficient and reliable.

Object Configuration Manager (OCM) settings

OCM is the J.D. Edwards vehicle for configuring the placement of tables, business functions, event rules, generic text, and media objects. The placement can be on any server on the network and even on the local PC. For details on how to plan and configure the OCM settings, refer to 4.4, “Object Configuration Manager (OCM)” on page 54.

The OCM settings can play an important role in affecting the OneWorld performance. The OCM setting should point to the current machine if possible, and then to another machine on the LAN, and lastly to an object across the WAN. However, the cost of replication must be weighed against the cost of placement of the object.

Processing options

The selection of processing options should be dictated by the business needs. However, excess processing can sometimes be eliminated by changing the processing options to accomplish the same task, but by using a different method.

Setting the job logging level as 4 0 *NOLIST

To reduce the overhead due to a joblog dump at job termination, all the profiles used by OneWorld should have the log level set to 4 00 *NOLIST, unless the application is deliberately in a debugging and testing status. The profiles include ONEWORLD, which normally runs the services, JDE or what is used for connections, QUSER or what is used for prestarted jobs, and all of the users.

Database indexes (access paths)

OneWorld software basically uses dynamic SQL for table (file) accesses. In many cases, data accesses can be improved by creating additional indexes. However, keep in mind, that each extra index causes another seize (system-instigated lock) while that index is being updated. Also, these indexes have to be maintained for every insert and update of indexes, which is an additional overhead. Therefore, before creating new indexes, ask yourself whether that index is a one time requirement or whether it is used multiple times. If a new index is created that can use an existing index, it will be implicitly shared automatically by the AS/400 database. For a one time requirement, in most cases, the AS/400 query optimizer automatically builds a temporary index. The suggested method to find permanent index candidates is to use Database Monitor tool. See 21.3.5.1, "Database Monitor statistics" on page 616, for more details on the Database Monitor tool.

Access path size

AS/400 indexes are defined as either a maximum 4 GB index or a maximum 1 TB index. The 1 TB index is relatively new and it is currently the default used when new indexes are created. In general, the 4 GB index is a little faster for smaller non-dynamic tables and the 1 TB is better for large dynamic tables. Another consideration is that the entire table is seized for the insertion of an index into a 4 GB index, while only the appropriate page is seized while inserting an entry into a 1 TB index. The new 1 TB indexes have another advantage in that they can be built in parallel if the system is a multi-processor system. For best performance, use the 1 TB index. Beginning with V4R1, the default is to use 1 TB indexes. However, customers migrating from earlier releases and installations may still have some 4GB indexes. We recommend that if a coexistence environment exists and that many of the WorldSoftware indexes are still 4 GB in size, all indexes should be changed in reverse alphabetic sequence using the 1 TB index option (ACCPHSIZ). This does not only optimize the index itself. However, by creating them in reverse alphabetic sequence, it also causes the primary key to be the first index considered by the query optimizer. This is because the query optimizer uses the latest index created as its first index to consider. To see if an index is 4 GB or 1 TB index, enter either of the following commands:

```
DSPPFD lib/Logical file name
DSPPFD lib/*all to list all the files in the library
```

The following entry shows a 4 GB index:

```
Access path size .....: ACCPHSIZ *MAX4GB
```

This can be changed to MAX1TB by entering the command:

```
CHGLF logical_file_name ACCPHSIZ *MAX1TB
```


Note

When an index type is changed, it is rebuilt. This can result in a long running process.

Access path sharing

If a new index is created that can use an existing access path, it will be implicitly shared automatically by the AS/400 database.

Reusing deleted records

When the AS/400 system deletes a record, it normally flags the record as deleted, but it does not reuse the space. This allows the file to maintain the relative record numbers intact. However, it slows down disk access if many records need to be read at the physical I/O level that do not get passed to the logical I/O level simply because they are deleted. Deleted records can be removed by reorganizing the physical file or copying the file with `COMPRESS (*YES)`. A better method is to change all of the physical files to reuse the space by changing the physical file to say `REUSEDLT (*YES)`. This is possible since J.D. Edwards does not use relative record number (RRN) processing in OneWorld. The impact on performance from implementing the reuse deleted records function varies depending on the type of operation being done. It is particularly important to use the reuse deleted record option for work files on a large system since the nature of a work file is to insert records and delete them after the task is successful.

Symmetric Multiprocessing (SMP) feature

The DB2 UDB for AS/400 SMP feature provides application transparent support for parallel query operations on multi-processor AS/400 systems. The database manager can automatically activate parallel query processing to engage one or more processors (CPU and I/O processors) to work simultaneously on a single query. The DB2 UDB for AS/400 SMP can provide significant performance improvement when sufficient CPU and memory resources are available on the AS/400 system. The degree of parallel processing can be set system wide by setting the system value `QQRVDEGREE` to one of the following options:

- ***MAX**: Number of tasks to use most or all CPU resources
- ***OPTIMIZE**: Number of tasks to use a good share of CPU resources
- ***NBRTASKS**: Number of tasks not to exceed specified number
- ***IO**: I/O parallelism only
- ***NONE**: No parallelism

The SMP feature can be used only if DB2 Symmetric Multiprocessing feature is installed on your AS/400 system. This feature provides the following query optimization algorithms:

- **Parallel table scan**: Provides parallel operations for queries requiring a sequential scan of the entire table. Multiple tasks are used to scan the same table concurrently. Each task performs selection and column processing on a table partition and returns selected records to the requester.
- **Parallel key selection**: Provides parallel index operations for key selection. Multiple tasks are used to scan the same index concurrently.
- **Parallel key positioning**: Multiple tasks are used to perform the key positioning on a partitioned subset of the key ranges. Only those index entries that are within a selected range are processed.

- **Parallel index build:** This can speed up the process of building a temporary index, when the query optimizer cannot locate an appropriate index. It also speeds up the process of building permanent indexes, particularly when built over large database files.
- **Parallel index maintenance:** This is used when SMP is installed and the degree attribute for the job (or system) is set to *NBRTASK, *OPTIMIZE, or *MAX, the application is performing blocked inserts (at least 8 rows per block), and at least two indexes are to be maintained.
- **Parallel data loader:** This is a new function in OS/400 V4R3 (also available via PTF on V4R2 and V4R1) that makes loading AS/400 database tables from external data much simpler and faster. The data loader can import fixed format, delimited, and byte stream files. See 19.2, “Data migration” on page 534, for more information. The following PTFs are required for V4R1 and V4R2 releases of OS/400:
 - V4R1M0: SF47138 and SF47177 for OS/400
 - V4R2M0: SF46911 and SF46976 for OS/400

If AS/400 resources are available (particularly the memory and CPU are not a constraint), changing the system value QQRYPDEGREE to *OPTIMIZE may provide a better performance for the OneWorld applications. Optionally, you may set the “multiprocessing degree” for specific jobs by using the CHGQRYA DEGREE (*OPTIMIZE/*MAX ...) command. For more information on the SMP feature, see “Enabling Parallel Processing” in Chapter 24, “DB2 UDB for AS/400 Data Management and Query Optimizer Tips”, of *DB2 UDB for AS/400 SQL Programming*, SC41-5611.

Parallel pre-fetch

Parallel pre-fetching of data can be invoked through the CHGQRYA command or by using the system value QQRYPDEGREE *IO. The support for this is part of the base operating system, so no additional software components are required to take advantage of this feature. Pre-fetching involves the reading of data from multiple disk arms in parallel and reduces any performance impact resulting from disk arm contention.

However, there is a memory overhead of approximately 1 MB per actuator. Selecting Expert Cache for the memory pool is a prerequisite. Also, the increased availability of data compresses the processing into a shortened elapsed time and increases the percentage of CPU utilization during execution.

Parallel pre-fetch requires Expert Cache to be “on”.

```

Change Query Attributes (CHGQRYA)

Type choices, press Enter.

Query processing time limit . . *NOMAX      0-2147352578, *NOMAX,
Parallel processing degree . . . *IO       *NONE, *ANY, *SAME

```

Figure 286. Change Query Attributes

‘Pin’ frequently accessed objects in memory

Selected database and program objects can be pinned in memory by using the SETOBJACC command. This can reduce the amount of physical disk accesses,

reduce overall disk arm and IOP utilization, and improve disk response time. For database objects, you can load just the data (data space) or the index (access path). In some cases, you can guess which files are candidates for SETOBJACC by knowing the application. The PEX Trace collection can also be used to identify SETOBJACC candidates. For details on how to identify SETOBJACC candidates through the use of PEX Trace, see Chapter 9 of *AS/400 Performance Explorer Tips and Techniques*, SG24-4781. You can download the V4R3 and V4R4 versions of the SMTRACE and QYPINT libraries from the Web site at: http://www.as400service.ibm.com/p_dir/pegui.nsf

A GUI tool, PEXGUI, which is available on this site, can also be used to identify files involving a large number of reads.

SETOBJACC typically can cause the biggest performance improvement for applications with heavy random access to database files. Sequential processing of the files may also benefit. However, in most cases, standard system support for sequential processing or the use of Expert Cache can provide sufficient data caching.

If you know which objects will be good candidates for placing in main storage, you need to perform the following two steps:

1. Allocate memory for the objects.
2. Use the SETOBJACC command to load the objects into the allocated memory.

Allocate memory

You must dedicate some main storage, or memory, to be used to contain the selected objects. This can be done by either creating a new subsystem description or modifying an existing subsystem description to allocate memory that will be used exclusively for containing the selected objects. Here is an example of the CL command to create a new subsystem called SETOBSBS, with a private pool that should be used exclusively for the selected objects:

```
CRTSBSDB SBSD(QSYS/SETOBSBS) POOLS((1 20000 1)) TEXT('Subsystem to hold
selected objects in main memory')
```

Using the SETOBJACC command to load the objects

First, you may want to make sure the pool is actually cleared. Initially it is, but you may want to clear it whenever you reload it. The CLRPOOL command does this, and it makes sure that all pages which are in the pool are written out to disk. An example of the CLRPOOL command is shown in Figure 287.

```

Clear Pool (CLRPOOL)

Type choices, press Enter.

Storage pool:
  Shared pool or subsystem name      SETOBSBS      Name, *JOB, *SHRPOOL1.
  Pool identifier . . . . .          1             1-10

```

Figure 287. Clear Pool

Next, you begin loading some objects into the storage pool in the subsystem SETOBSBS. From an AS/400 Command Entry display, with F10 (Include detailed messages) enabled, load the selected objects (one at a time) as shown in Figure 288 on page 592.

```

Set Object Access (SETOBJACC)

Type choices, press Enter.

Object . . . . . Fxxxx      Name
Library . . . . . OWLIB      Name, *LIBL,
Object type . . . . . *FILE      *FILE, *PGM
Storage pool:
Shared pool or subsystem name      SETOBSBS      Name, *JOB, *
Pool identifier . . . . . 1      1-10

```

Figure 288. Set Object Access

The database objects loaded into memory must be selected with a good understanding of the application and database.

When loading an object into the specified pool, you *must* ensure the entire object fits into the pool. If only part of the object fits, you may experience poorer performance than before using the SETOBJACC command.

Some other important factors to keep in mind when deciding to use the SETOBJACC command are:

- Deleted record space is loaded into the main storage. Consider reorganizing the files (use the RGZPFM command) or re-using deleted record space before “pinning” a file in memory.
- SETOBJACC is “pool specific”. Once the command is issued for an object, the object stays in the pool until you clear the pool, issue another SETOBJACC command, or specify POOL(*PURGE) for the object.
- If you specify loading the same object in the same pool again, the previous copy of the object is purged and the object is reloaded.

Journaling

The primary purpose of journal management is to provide a method to recover database files. Additional uses related to performance include using journaling to decrease the time required to backup database files and using access path journaling for a potentially large reduction in the length of abnormal IPLs. For more information on the uses of journal management, refer to *AS/400 Backup and Recovery*, SC41-5304.

The addition of journaling to OneWorld applications impacts performance in terms of both CPU and I/Os. Also, the job that is making the change to the file must wait for the journal I/O to be written to the disk, so the response time will, in many cases, be affected as well. Journaling impacts the performance of each job differently, depending largely on the amount of database writes being done.

The impact to performance from adding journaling can be reduced by locating the journal receiver on a separate user auxiliary storage pool (ASP) that has an adequate number of disk arms. This results in an improvement in response time and in overall system throughput.

Commitment control

Commitment control is an extension to the journal function that allows users to ensure that all changes to a transaction are either all complete or, if not complete,

can be automatically backed out. Using commitment control adds one more journal entry at the beginning of the transaction in addition to the one added at the end of the transaction. This results in additional CPU and I/O overhead. In addition, the time that record level locks are held increases with using commitment control. Because of this additional overhead and possible additional record lock contention, adding commitment control may result in a noticeable degradation in performance of an application that is currently performing journaling.

However, there are times where adding commitment control can result in improved performance. Journaling, alone, means that the journal entries for a change to a file (table) are written synchronously to disk. However, under commitment control, most journal entries are written to disk asynchronously. Only the final journal entry of the commit cycle (along with any entries of the cycle that have not yet been written to disk) are written synchronously. Because of this, applications may no longer have to wait for each additional journaled change to be written.

It is important to remember that the potential for improving response time by adding commitment control is largely affected by overall system resource utilization. Systems that have constrained memory or high CPU utilization, in most cases, show degradation in performance. Also, adding commitment control can result in record level lock contention between jobs, which can also affect response time. Given the number of variables involved, a test run is highly recommended prior to adding commitment control for the purpose of improving performance in a production environment.

For an explanation of how OneWorld implements Commitment Control and Journaling, see 6.6, "J.D. Edwards database implementation" on page 98.

Transaction processing

Enforcing transaction boundaries causes a significant amount of work. It also drastically increases lock contention problems. This cost needs to be considered. If transaction processing is to be turned off, it is done so at the individual application level.

Reducing the search criteria

One of the easiest means of increasing performance is to educate the users to narrow their search criteria. For example, if only a single character is used as the search element, the application may have to search a large set of data causing poor performance. In contrast, if the search is done with a fuller selection criteria, fewer records are searched which results in much better performance.

UBE performance tuning

UBE jobs normally run at a lower priority of 50. The performance of these jobs depends on a number of factors, such as:

- Available CPU at UBE job priority
- Number of AS/400 processors
- Number of batch threads
- UBE run priority
- Size of memory in the batch pools
- Disk response time
- Data organization in the files being processed by the batch jobs
- SQL query optimization

- Use of Query Degree and the SMP feature
- Use of journaling or commitment control

CPU

If CPU utilization is too high at the UBE job priority, these jobs may not be able to get enough CPU cycles to complete within the available window. In a situation where CPU is a constraint for batch processing, one of the following corrective actions may be required:

- Upgrade the AS/400 server.
- Reduce the number of batch threads.
- Increase the priority of critical UBE jobs.

A good rule for n-way AS/400 systems is to limit the number of concurrent UBE jobs to n-1. This ensures that there is always at least one processor free to service user requests. Also, on a uniprocessor AS/400 system, run the UBEs at a very low priority (for example, 95), and set their time slice to 50 ms. If you plan to perform a lot of UBE processing, consider using a multiprocessor AS/400 system.

Memory

Memory plays a critical role in the processing of long running SQL queries. The SQL queries tend to run faster when a lot of memory is given to their pool. In general, give as much memory as possible to the UBE pool during the processing of long running SQL queries. SQL queries performing activities such as scanning large tables, using hashing algorithm for grouping or joining, and sorting data in a large table can take advantage of larger memory. This is specially true when the SMP feature is in use.

Disk and I/O performance

In general, batch jobs are I/O intensive in nature. Improving the disk performance and reducing the number of physical accesses (I/Os) can play a key role in improving the performance of batch jobs. The following recommendations can help reduce the physical I/Os and improving disk response:

- **Reorganize physical files in the order of sequentially accessed key**

Even with the use of Reuse Deleted Space feature, it may still be necessary to periodically reorganize physical file members. The reorganization process removes deleted record space and can also arrange the data physically in the order of the keys specified within a logical file. This can reduce the number of physical I/Os required when a file is processed sequentially by key, since the next requested physical record may already be in the memory.

Attention

Reorganizing a physical file member may force the rebuilding of access paths. This may cause performance slowdown. Therefore, this should be done in an off-peak period.

Use SETOBJACC for selected files for critical UBE job processing.

This was already explained in “‘Pin’ frequently accessed objects in memory” on page 590.

- **Use Expert Cache for the UBE pools**

Please refer to “Turning ‘ON’ Expert Cache” on page 578.

- **Minimize file journaling for UBE processing**

If use journaling, create a separate ASP for the journal receivers. Also, using commitment control for UBE processing provides better performance than using journaling only. For more details, refer to “Journaling” on page 592, and “Commitment control” on page 592.

- **Use SMP feature for jobs involving SQL processing**

Using the `CHGQRYA *OPTIMIZE` command for jobs involving long running SQL queries may provide much better performance. For more details, refer to “Symmetric Multiprocessing (SMP) feature” on page 589.

- **Tune the SQL statements by using DBMON or STRDBG tools**

Optimize the SQL queries used in the UBE jobs. In many cases, creating proper indexes or modifying the query syntax than a different selection can provide much better performance for the UBE jobs.

21.2.3.5 Configuring SQL package libraries

SQL packages are a feature of the AS/400 system that store information about dynamic SQL queries on the AS/400 system. That way, the next time the statement is executed, the query optimizer already knows the best method to use to execute the query. In addition, if the system detects significant changes in the database, it re-optimizes the query and saves the updated query information.

There is a package for each application. The package name is the first seven characters of the application, plus a three-character hash. For example, typically OneWorld generates a package name of OEXPLORFJA.

This package is a file that is stored in a library on the AS/400 system. The default library in which the packages are stored is QGPL.

In addition, associated with each package is a default library. This default library is defined in the ODBC configuration. It is the first library listed under Default libraries of the Server tab in the ODBC configuration screens. For the purpose of this section, we refer to this library as the DefaultDataLibrary to distinguish it from other “default library” terms.

If an application uses multiple ODBC Data Sources and they have a different DefaultDataLibrary, the result is that only the first data source that was referenced will use the SQL packages. This is because when a data source is used that has a different ODBC DefaultDataLibrary, it will not store SQL information in the package. The net result is that for these data sources, queries will take longer because the method must be determined each time it is executed.

The way to avoid this problem is to define a different library in which to store the SQL package for each data source that has a different DefaultDataLibrary. Table 45 illustrates this.

Table 45. Same default library is used for all SQL packages

DatasourceName	DefaultDataLibrary	Library for SQL package
Data1	PRODDTA	QGPL
Data2	PRODCTL	QGPL
Data3	B732CTL	QGPL

DatasourceName	DefaultDataLibrary	Library for SQL package
Data4	PRODDTA	PRODDTA

In the example in Table 45, if the first time the application is executed, Data1 is the first data source used. Then, the library associated with the package is PRODDTA. Any statements for Data1 and Data4 are stored in the package because they both have the same default library. However, for statements against Data2 and Data3, no statements will be stored. Therefore, these queries will take longer because the method needs to be determined each time the query is run.

Change the definitions to ensure SQL packages are used for all data sources.

Table 46. SQL package library is the same as the DefaultDataLibrary

DatasourceName	DefaultDataLibrary	Library for SQL package
Data1	PRODDTA	PRODDTA
Data2	PRODCTL	PRODCTL
Data3	B732CTL	B732CTL
Data4	PRODDTA	PRODDTA

In this definition, the SQL package is put in the same library as the DefaultDataLibrary.

In Table 47, different SQL package libraries are used for the same DefaultDataLibrary, PRODDTA.

Table 47. Different SQL package libraries for the same DefaultDataLibrary

DatasourceName	DefaultDataLibrary	Library for SQL package
Data1	PRODDTA	DATA1
Data2	PRODCTL	DATA2
Data3	B732CTL	DATA3
Data4	PRODDTA	DATA4

In this case, a new library is defined (must be created on the AS/400 system) to store the SQL packages for each data source. In this case, even though both Data1 and Data4 have the same DefaultDataLibrary, the SQL packages will be stored in different libraries. With V4R3, searching is done with hashing and is very fast.

The key point is that for ODBC Data Sources, the SQL package only has one DefaultDataLibrary associated with it. Therefore, to ensure that SQL packages are used, all ODBC Data Sources need to be configured so that the library specified for the SQL package have the same DefaultDataLibrary.

One way to determine if there is a possible problem in the library being defined incorrectly for an SQL package is to turn on the SQL trace function on the client, run the application, stop tracing, and then look at the log file created.

If there is a problem with the SQL package, a warning message is shown in the log indicating: `Extended Dynamic has been disabled`. This is a strong indication that the `DefaultDataLibrary` is not the same as the SQL package file that this connection is using.

Configuring separate SQL packages by library

Typically these values are set on the Deployment Server in the `ODBCDataSource.inf` file. However, should you need to edit a particular client PC, it can be done by entering the library to use for the SQL packages for each data source on the Package(s) tab of the ODBC Administrator for Client Access Express.

After updating the SQL Package library, you should delete the old packages, so that they can be automatically recreated when the application is run again.

To find out what SQL packages exist on the AS/400 system, enter the following command:

```
WRKOBJ OBJ(*ALL/*ALL) OBJTYPE(*SQLPKG)
```

Do not delete any system packages in libraries such as `QIWS` or `QSYS`.

You can also print SQL package information using the `PRTSQLINF` command. This is explained in 21.3.5.3, “Print SQL Information” on page 621.

For additional information, refer to *Client Access for Windows 95 ODBC User's Guide*, SC41-3535.

21.3 Performance analysis

This section focuses on the process of gathering performance-related data and using available tools to analyze that data to help solve performance problems. However, we begin by providing an overview and a prioritized list of “heavy hitters”, which are items that you need to consider first when you are having a performance problem.

21.3.1 Fixing performance problems: First steps

Understanding the performance problem in customer terms is key to minimizing the unnecessary analysis of low priority problems or unrealistic customer expectations, such as an AS/400 20S server model handling 200 busy ODBC client workstations.

The methodology for investigation involves understanding the perceived problem, including:

- Application function
- Design overview
- Expected response or throughput
- Perceived response or throughput

It is important to understand the problem as perceived by the users. The expected response time goals need to be established and validated against the application architecture to ensure that it achievable. Such questions that are asked include:

- Is the problem system-wide, or just for a specific application?
- If it is for a specific application, is the problem just for certain transactions and not others?
- Does the problem only occur during some specific time periods during the work day?

The user's locality to the server is also a consideration in this regard, since line time is significant if the user is accessing the server through a low bandwidth line or heavily utilized line.

Review the overall system performance with consideration to the following areas:

- Hardware error log
- System values
- Page faulting
- Activity levels
- Disk arm activity
- Disk space usage

Examine the system to ensure that the system parameters affecting performance are acceptable and that memory pool allocation and activity levels are adjusted to suit the system. This preliminary review is done by checking the system interactively. The purpose of this is to avoid getting deeply involved in an application performance investigation when the problem may be with the setting of system parameters. That is, it is a system resource problem, rather than an application design or implementation problem.

Carry out a detailed system performance or capacity study based on OS/400 Performance Monitor summary data or repetitive usage of system commands, such as Work with System Status (WRKSYSSTS) and Work with Disk Status. Using the commands to see important information, requires significant human intervention to gather sufficient data over a meaningful period of time. Collection of Performance Monitor data and analysis of that data through the Performance Tools/400 Advisor function and individual printed reports provide a more comprehensive set of information.

Once the review of summary level data is complete, you may determine that system-level tuning can resolve the performance problem.

Using the hardware performance guidelines, you may determine that a faster CPU, more main storage, or faster or additional disk hardware is required.

Analyze the client/server application, in detail, by collecting detailed information while running the jobs in question.

Generally the faster the resource is, the higher the utilization threshold tolerance is. The slower the resource is, the lower the threshold tolerance is.

If utilization guidelines for the various resources are exceeded, you need to first resolve these issues. If performance problems exist and all resource measurements are significantly below the guidelines, application transaction performance data must be collected and analyzed by a performance specialist.

Note that Performance Monitor data collects and the Performance Tools/400 Manager feature reports 5250-based interactive response times. Even though

OLTP client/server environments are considered “interactive” from the end user's viewpoint, the AS/400 summary level performance data does not identify client/server transactions. Therefore, it is not able to calculate and report “non-interactive response time”.

OneWorld, like other client/server ERPs, extensively uses SQL queries. If they are not optimized for performance, these queries can severely degrade the server performance. Tools such as STRDBMON should be regularly used to analyze database and SQL performance. Another performance issue is that of creating the right number access paths on the AS/400 system. Analyzing the available access paths through the use of such commands as Display Database Relationships (DSPDBR), creating the required indexes, and deleting the unnecessary indexes on regular basis should be done.

Use the following series of steps to attack specific performance problems. They are listed in order from the most important cause of performance problems to the least important cause, in our experience.

Remember

- Look first at the big picture.
- Don't jump in.
- Do most of the steps in sequence.
- Doing the steps out of sequence causes many lower priority steps to be redone.

1. **Network design:** Make sure that your topology is configured for optimal application performance. See Chapter 5, “OneWorld implementation designs” on page 71, for examples and considerations for different topology implementations.

2. **System:** Look at the system from a very high-level perspective:

```
Estimated CPW requirement = (OneWorld Users) * (3 to 40)
+ (World Users) * .718
+ "Other" workload
```

As a rule of thumb, OneWorld requires approximately five times the CPW an existing World implementation requires to deliver comparable performance.

If this high-level assessment indicates the machine is seriously undersized, your efforts should focus first on validating that the hardware is overused and, if necessary, upgrading the equipment to a reasonable configuration.

3. **Disk hardware:** Look at the disk hardware configuration. It is important that there is a sufficient number of arms. In our experience, configurations utilizing 17 GB drives rarely have enough arms to accommodate OneWorld disk I/O requirements. We recommend using 4 GB drives, with 8 GB drives being acceptable.

Also, make sure that there are enough disk IOPs and that the drives are evenly distributed across the available IOPs. With current IOPs, it is acceptable to attach up to 15 drives per IOP, except for the MFIO, which should be limited to 10 drives.

Make sure that the drives are less than 80% full. When space usage exceeds this threshold, disk activity related to extent allocation may increase to the point that performance suffers.

4. **Software versions:** Make sure that you are running on the latest available versions and service levels of OS/400, Client Access Express, and OneWorld. Both IBM and J.D. Edwards are constantly improving performance by incorporating lessons learned in customer installations into their respective software products. Refer to 22.2.1, “Informational APARs” on page 717, for a list of OneWorld related AS/400 APARs and how to obtain them.
5. **Disk software:** Make sure that you are using your existing disk configuration most efficiently:
 - Are the journal receivers in a separate ASP? Journal performance is best if the heads used for writing journal entries are dedicated.
 - Does the separate ASP have enough drives? Journaling can use up to 13 drives in round-robin fashion for best performance.
 - Is the separate ASP mirrored or RAID protected? It is best to mirror for performance.
 - Are there multiple receivers in the same ASP? Much of the benefit of a separate ASP is lost if the heads have to move between multiple receivers.
 - Are the disk drives evenly used? If new drives have been added, you may need to redistribute the data across all the drives now on the system.
 - Is replication activity affecting performance? If the journal receivers are being read frequently, head movement again becomes a factor. Also, if replicating across a network, replication data traffic may be affecting network performance.
6. **System tuning:** Make sure that the system is using its existing resources efficiently:
 - Is the Machine Pool fault rate within recommended guidelines (see *OS/400 Work Management Guide*, SC41-5306)? Excessive faulting in the Machine Pool affects all jobs on the system.
 - Are the fault rates in other memory pools within recommended guidelines? Again, excessive faulting, especially in a pool being used by OneWorld, will affect application performance.
 - Are you mixing workloads (for example, batch and interactive) in the same pool? Different workload types should be segregated into different memory pools so that their different execution characteristics don't interfere with one another.
 - Are you experiencing an excessive amount of interactive utilization on a server system? On these systems, even moderate amounts of interactive activity can adversely affect batch performance.
7. **Normal-mode operations:** Make sure that the system is returned to normal-mode operations after a problem diagnosis session:
 - a. Turn off job logging for QZDASOINIT jobs.
 - b. Set the MAXUSE parameter of the application's prestart job entries to the recommended value of 200.
 - c. Make sure the accounting level, audit level, JDE Debug on both Client and Server, and ODBC trace settings are appropriate for normal operations. Unnecessary logging can contribute significantly to poor performance of both interactive and batch (UBE) jobs.

8. DB2 UDB for AS/400 tuning: Make sure that DB2 is working efficiently:

- Are the right indexes in place? Check for logical files missing from the standard set, or large numbers of superfluous indexes. Use DBMON, described in 21.3.5, “AS/400 database query optimization tools” on page 616, to identify good candidates for new indexes.
- Are SQL packages working efficiently? If the access plans currently being used by the packages are not optimal, it may be necessary to delete the packages and let the system rebuild them.
- Are there any relatively small, high I/O files that could be made memory-resident using SETOBJACC?
- Are there any files that are being unnecessarily journaled?
- Are the indexes over large, heavily updated files Terabyte indexes rather than the older 4 GB indexes? These newer indexes have less contention because they are locked during index update operations at the page level rather than at the table level.
- Are there any indexes that are redundant, especially cases where the indexes are identical except one is a 4 GB index and one is a 1 TB index? In this situation, the index with the correct record name should be saved and changed to 1 TB and the other index deleted.
- Are there files with a large percentage of deleted records, especially work files? During processing, deleted records still have to be handled by the operating system, so it may improve performance if files with a lot of deleted records are periodically reorganized, or have their Reuse Deleted Records attribute set to *Yes.
- Are there work tables with garbage left behind by aborted transactions? If all work has been posted, it is usually possible to clear work tables, since any data in them is due to aborted transactions.
- Are there unnecessary Unique constraints in place for one or more files? The presence of a Unique constraint forces many file operations to be done synchronously, hurting performance, rather than asynchronously.

9. Excessive job waits: Make sure that system resources are made available to jobs, within the limits of the system’s capacity, to avoid having jobs wait for resources unnecessarily.

- Are there excessive seize (typically, for index updates) or lock (typically, record lock) waits? Making sure that 1 TB indexes are used, as discussed previously, is one way to reduce seize waits. Making sure that error messages are not going unanswered is one way to reduce record lock waits, particularly for the next-numbers tables, F0002 and F00021.
- Are enough net processes running? If, on a given server, the maximum number of connections for the JDENET_n job is met, a client or server cannot initiate an additional OneWorld session on that server until an existing session connection is ended.
- Are enough kernel processes running? Kernel processes are deliberately round robined to perform load balancing. However, a user is allocated to a particular kernel process the first time the user requests that process. If another user consumes the kernel then the requestor must wait. B732 fat client testing showed that the call process kernel could handle up to 10 users on a LAN or up to 20 users on a WAN. However, these tests were

done using Windows NT where memory is more of a problem. One kernel per user may make sense on the AS/400 system since the unused kernels will simply be paged out.

- Is a single, long-running business function consuming the kernel?
- Is a Universal Batch Engine (UBE) process waiting in a job queue?
- Is an AS/400 job waiting for an Activity Level in the subsystem?

10. **Workload reduction:** Are you asking the system to do unnecessary work?

- Has history data been purged to keep file sizes reasonable?
- Can users request information in a more efficient manner?

11. **OneWorld setup:** Make sure that all OneWorld configuration settings have been properly specified:

- Are all server names valid (both active and commented out)? OneWorld has used the server name in the INI file for minor purposes other than the obvious reason. If this occurs and the server name is invalid, the entire network is searched attempting to perform name resolution.
- Is the OCM configured for efficiency? OCM defines the placement of the tables. If the table is available locally, the OCM should reflect this. See the OneWorld guide *Configurable Network Computing Implementation* for recommendations.
- Is row security performed by SQL (versus the application)? Row security was originally implemented by the application. This was switched to using the DB implementation for performance. If the setting still exists in the INI file, it should be removed.
- Are the static tables cached to the client? Tables that are static but highly used for a OneWorld implementation should be defined in the F98613 table so that they are cached to the client. Some of the common examples are:
 - F03B25: A/R delinquency policy table
 - F98950: User overrides table
 - F98980: Font override by language
- Consider turning off QBE for non-indexed key fields. The QBE line can be used for any field displayed. If an index does not exist for the field selected, performance will be slower. Since indexes do tend to exist for the common search fields, this additional capability can be turned off to prevent unnecessary overhead.
- Is PrintImmediate set to False and printer type to *line? The fastest definition of printing on the AS/400 system is *line. Another question is, do you really want AS/400 spooled files for all UBEs submitted as well as PDF files. If not, change PrintImmediate to FALSE.
- Are UBE specs pushed to the server every time? If the UBE specs are stable, it does not make sense to push the specs to the server every time the report runs. This can be changed in the INI file section [Network Queue Settings] using InstallSpecs.
- Are the appropriate ESU and application paper fixes applied? If performance is a problem for a particular application, the response line should be contacted since frequently a paper fix can resolve the problem.

- Was the server package built using optimize(*basic)? If not, you can run the CHGSRVPGM SRVPGM(*ALL) OPTIMIZE(*BASIC) command against the application library to optimize the service programs for better performance.

12. **Additional system tuning:** Is the AS/400 system reasonably tuned? Some additional tuning parameters to consider are:

- How many concurrent batch jobs execute system wide? This should be limited by the number of jobs that can simultaneously come from each job queue. It should also be limited by the number of jobs that can concurrently execute within the subsystem. If the AS/400 system has only 1 CPU, the time slice for QBATCH is frequently reduced to 500ms to allow user requests to be processed more quickly when a UBE is active.
- How many OneWorld batch job threads? OneWorld batch jobs can be set up to run from separate job queues based upon major application. In addition, G/L posting job R09801 is frequently set up to run from its own single threaded job queue.
- QSERVER class settings
 - Purge(*NO). Recent tests have shown that the QSERVER class should be left at default value for purge (*NO). This is a change based on tests performed by IBM.
 - Time slice
- Prestart jobs:
 - Number of prestart jobs: The number of prestart jobs should be based upon usage over a period of time. If too many jobs are defined, the subsystem will take a long time to shutdown. If too few jobs are defined, dynamic invocation of the jobs will slow down the system.
 - Start at subsystem start: If the jobs do not start at subsystem startup, the first use will be significantly slower than subsequent uses.
 - Reuse prestart settings: The prestart job QZDASOINIT should normally be set to reuse for the default of 200 uses to minimize the overhead of job creation and shutdown. However, if stability is a problem or for debugging, it should be set to 1. This will cause the logs to be written for each usage.
- Automatic performance adjust turned off. If the system is only used for OneWorld, the automatic performance adjustment should be turned off so that pool size does not change. Constant changing of the pool size causes SQL to constantly rebuild the access plan.
- Use shared pools. Shared memory pools is the best means of combining similar work in a common pool. Use the WRKSHRPOOL command to set the size and activity level, and to set the Paging Option to *CALC.
 - Fixed in size
 - Memory available for SQL execution = Pool size/Activity level
- Over-managed subsystems, pools, etc., it is common to see AS/400 systems that are over managed. At one point in time, the tuning was correct, but frequently the tuning is not changed as usage of the system changes.
- Library list reasonable in job descriptions

13. Network setup: Is the communications network setup efficient?

- Are the ping times reasonable? Ping times on the local network should be less than 10 ms on the AS/400 system. It is helpful to write a ping program to verify that this is consistently true. If the ping times have periodic spikes, this may mean that the network card is overloaded. One possible cause of this is interactive UBEs.
- Multiple NIC cards? Multiple NIC cards are now automatically fully supported. There are no setting required in the INI file.
- Duplex setting on line description properly set? A mismatch of duplex on the network frequently occurs. A method of testing this is to copy a large file from the AS/400 IFS directory to a client and back. If the times differ, it is likely a mismatch in the AS/400 line description versus either the router or the hub configuration.
- Any NIC card problems? Intermittent problems with NIC cards cause symptoms similar to duplex mismatch.
- Is DNS name resolution efficient?

14. ODBC settings: Are the latest recommendations for ODBC settings being observed?

- Package support
 - Enable extended dynamic support
 - Separate library for each data source
- Cache package locally
- Lazy close checked
- Prefetch on execute not checked
- Record blocking type = Block for update of
- Record blocking size = 8
- OS/400 library view = Default library list
- Bypass sign on via registry (Unnecessary for CA express)

15. Job management: Is the number of jobs being managed by the system kept to a reasonable number, and are job-related system values set appropriately for the level of activity on the system?

- How many jobs exist? If cleanup does not happen on the AS/400 system, internal table sizes will grow and performance will slightly suffer. On the AS/400 system, the job is entered into the table when it enters the job queue and leaves the system when the last spooled file entry is deleted.
 - Spool files
 - Job logs
 - History logs
- Do system values reflect the number of jobs? QACTJOB and QTOTJOB establish the initial values and QADLACTJ and QADLTOTJ are the increments for expansion for the total number of jobs and the total number of active jobs on the system, respectively.
 - QACTJOB
 - QTOTJOB

- QADLACTJ
- QADLTOTJ

16. **Network parameters:** Are AS/400 communications parameters optimized?

- Max transmission unit on AS/400 TCP/IP set to *IFC
- TCP/IP buffer size set to 64K
- Communications line utilization under 20%

21.3.2 Performance analysis tools

The following tools can be used for performance analysis:

- **AS/400 server tools**

- AS/400 Performance tools Reports
 - Advisor (display or printed output)
 - System
 - Component
 - Transaction Summary (requires Trace *ALL option)
 - Transaction Detail (specific jobs only - requires Trace *ALL option)
 - Lock Report (requires Trace *ALL option)
- Database Query Optimization Tools such as Database Monitor, Debug Messages and Print SQL information.
- AS/400 Performance Explorer (PEX) *STATS can be used to find resource intensive programs/ SQL module for a selected job or for the entire system.
- AS/400 PEX *TRACE can be used to identify I/O intensive objects and SETOBJACC candidates.
- Job Trace.
- For system-wide database analysis, commands such as DSPFD, WRKJRNA, and DSPDBR may be used.

- **Client Access Express and Client application PC tools**

- ODBC trace

In most cases, collection of the ODBC API trace data results in a significant performance degradation on the client workstation. We recommend that you collect ODBC API trace information at a time separate from collecting the AS/400 server performance information.
- JDEDebug.LOG
- Server Administration Workbench (SAW)
- DBLOG and ODBCLOG

- **Communication Analysis Tools**

- Communications trace
- Sniffer and other vendor tools such as Optimal, CACI, and MIL3

21.3.3 AS/400 server performance measurement: Regular data collection

The performance data collection procedure for the following tools is covered in this section:

- **Performance Monitor data:** To determine resource bottlenecks, if any
- **Database Monitor data:** To find database or SQL tuning opportunities

Performance data may be collected from these tools on a regular basis. Data collected from these tools can be analyzed to find bottlenecks related to AS/400 hardware resources, AS/400 work management tuning, and database SQL tuning. Many other performance tools for the AS/400 system and OneWorld tuning are available. They may be used as needed, depending on the nature of the performance problem. A complete set of performance analysis tools is mentioned in the previous section. A number of data collection procedures involving these and some other tools are provided in 21.4, “AS/400 performance data collection procedures” on page 663. You may also consider using a third-party tool called DB Essentials from Centerfold. This tool has a user-friendly interface for Database Monitor data collection and analysis.

21.3.3.1 Collecting Performance Monitor data

The AS/400 system has a standard performance data collection facility that is initiated by using the STRPFRMON command that is part of OS/400. The files created by this command are described in the *OS/400 Work Management Guide*, SC41-5306, and users can write their own queries to analyze the data. Performance Monitor files all start with the QAPMxxxx prefix. However, we recommend using the Performance Tools for AS/400 (5769-PT1) Manager Feature to print the reports and run the Advisor.

In most situations, the Performance Monitor is started with TRACE(*NONE). No job level trace data is collected. This is acceptable as the first step is to analyze performance from a system level. However, we suggest that you initially use the trace option (TRACE(*ALL)) since it saves having to re-run the application again if it seems system level performance information indicates no bottlenecks and application analysis is required. The following sections show how to collect Performance Monitor data. After the data is collected, you should run the Performance Tools/400 Advisor against the collected data. Reviewing the Advisor messages may direct you to specific system tuning actions. Assuming you need to further analyze the collected Performance Monitor data, you can print the various Performance Tools/400 reports. Performance data with the trace option provides detailed information of activity on the AS/400 system at the application level. In general, the System Report and Component Reports use system level or “sample data” for their output, system level information. Transaction and Transition Reports use trace data and are intended for application level review.

Starting Performance Monitor

Performance data can be collected on the AS/400 system with the STRPFRMON command. Type STRPFRMON on the AS/400 command line and press F4 to view the display shown in Figure 289.

```

Start Performance Monitor (STRPFRMON)

Type choices, press Enter.

Member . . . . . MBR          OW10AM
Library . . . . . LIB          JDE_OW
Text 'description' . . . . . TEXT      Trace collection of ONEWORLD application
Time interval (in minutes) . . . . . INTERVAL      5
Stops data collection . . . . . ENDTYPE      *ELAPSED
Days from current day . . . . . DAY          0
Hour . . . . . HOUR            1
Minutes . . . . . MINUTE       0
Data type . . . . . DATA      *ALL
Select jobs . . . . . SLTJOB    *ALL
Trace type . . . . . TRACE     *all
Dump the trace . . . . . DMPTRC *YES
Job trace interval . . . . . JOBTRCITV .5
Job types . . . . . JOBTYP     *DFT
Start database monitor . . . . . STRDBMON *NO

More...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 289. Start Performance Monitor display

You may have to create a new library for your data. Ensure that the Trace type parameter is *ALL, Dump the Trace is *YES. The STRDBMON command is specified as *NO in this example. To collect the Database Monitor data along with the Performance Monitor, this may be selected as *YES. This collects data for one hour (HOUR = 1) and then ends the Performance Monitor. Dumping the trace data defaults to occur when the ENDPFRMON command is issued or when the Performance Monitor ends. On a very busy system, dumping the trace data during production mode can degrade normal system performance. You can optionally dump the trace at a later time (not recommended unless the system is too busy) by specifying DMPTRC *NO in the STRPFRMON command and by using the DMPTRC command whenever you want to dump the trace.

21.3.3.2 Collecting Database Monitor data

The Database Monitor (DBMON) collects valuable information for SQL and other query optimization. The DBMON data can be collected either by using the STRDBMON command as described in the following section, or by using parameter STRDBMON(*YES) in the STRPFRMON command. When collected from STRPFRMON command, the DBMON data is collected in the QAPMDBMON file.

Starting Database Monitor

To collect the Database Monitor data, enter `STRDBMON` and press F4. Specify the library, file, and member name for the file to collect DBMON data. Also, specify *SUMMARY for Type of Records and press Enter.

```

Start Database Monitor (STRDBMON)

Type choices, press Enter.

File to receive output . . . . . > DBMON          Name
Library . . . . . > DEMONCOLL      Name, *LIBL, *CURLIB
Output member options:
Member to receive output . . . > MON10AM        Name, *FIRST
Replace or add records . . . . . *REPLACE      *REPLACE, *ADD
Job name . . . . . > *ALL          Name, *, *ALL
User . . . . .                          Name
Number . . . . .                          000000-999999
Type of records . . . . . *SUMMARY      *SUMMARY, *DETAIL
Force record write . . . . . *CALC        0-32767, *CALC
Comment . . . . . Monday morning collection

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 290. Start Database Monitor display

The STRDBMON command starts the collection of database performance statistics for a specific job or all jobs on the system. For monitoring jobs *not* using non-SQL queries, use *SUMMARY for the type of records. For the first few iterations, data collection for all the jobs is recommended. After the database is tuned to a reasonable level of performance, you may focus on specific SQL queries by collecting data for specific jobs. The statistics are placed in an output database file and member specified on the command. If the output file or member does not exist, one is created based upon the file format definition of model file QSYS/QAQQDBMN.

You can specify a replace or append option that allows you to clear the member information before writing records or to just append new information to the end of the existing file.

Ending the Database Monitor

To end the Database Monitor, use the ENDDEMON *ALL command if you started it with *ALL.

21.3.4 Server performance analysis using Performance Tools/400

This section describes performance analysis and reporting using Performance Tools for AS/400 program product (5769-PT1). The other AS/400 performance analysis tools are described in the following two sections.

21.3.4.1 The Advisor function

The Advisor analyzes the Performance Monitor data and produces a set of messages under Recommendations and Conclusions headings. The Performance Tools/400 main menu is shown in Figure 291.

To get the advisor messages, enter the command:

```
GO PERFORM
```

```

PERFORM                      IBM Performance Tools for AS/400
                                System:  ASM23

Select one of the following:

    1. Select type of status
    2. Collect performance data
    3. Print performance report
    4. Capacity planning/modeling
    5. Performance utilities
    6. Configure and manage tools
    7. Display performance data
    8. System activity
    9. Performance graphics
   10. Advisor

    70. Related commands

Selection or command
===>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=Information Assistant
F16=System main menu

```

Figure 291. Performance Tools/400 main menu

When you select the Advisor option, a list of Performance Monitor data collections in the specified library appears as shown in Figure 292.

```

                                Select Member for Analysis

Library . . . .  JDE_OW

Type option, press Enter.
  1=Select  5=Display

Option  Member      Text                               Date      Time
-----  -
      OW10AM      Performance test          02/19/99  15:11:11
      FRIDAY02          11/06/98  17:00:21
      FRIDAY01          11/06/98  16:21:17

                                                                Bottom

F3=Exit  F12=Cancel  F15=Sort by name  F16=Sort by text
F19=Sort by date/time

```

Figure 292. Performance advisor option

After selecting the performance member, you are presented with a list of summary performance statistics for time intervals. You may select all intervals to be analyzed or a subset of time intervals. The Advisor analyzes the data collected and produces a set of messages under Recommendations and Conclusions headings. An example of the first-level message output is shown in Figure 293 on page 610.

```

                                Display Recommendations
                                System:  ASM23
Member . . . . . : OW10AM      Library . . . . . : JDE_OW
System . . . . . : RCHASPT2    Version/Release . . : 4/ 3.0
Start date . . . . : 02/19/99   Model . . . . . : 530-2151
Start time . . . . . : 15:11:11  Serial number . . . : 10-122BM
QPFRAJ . . . . . : 3           QDYNPTYSCD . . . . : 1

Type options, press Enter.
  5=Display details

Option  Recommendations and conclusions
      Recommendations
      Increase pool size for listed pools.
      ASP space capacity exceeded guideline of 80.0%.
      Conclusions
      Pools may possibly be removed.
      Pool fault rates approached guideline.
      Pool fault rates below guideline.
      Pool fault rate zero in all intervals.
      Disk seek length exceeded guideline.

More...

F3=Exit  F6=Print  F9=Tune system  F12=Cancel  F21=Command line

```

Figure 293. Performance advisor recommendations

We recommend that you use the Advisor as an initial view of overall system performance data. Review the message details to determine what additional performance analysis should be performed. You may also invoke the Advisor function and output by using the Analyze Performance Data (ANZPFRDTA) command.

The System Report and Component Report contain much more overall system performance information. Reviewing Advisor output can often speed up further analysis of the Performance Tools reports.

21.3.4.2 Performance Tools/400 reports

Enter the `GO PERFORM` command, and enter option 3 to print performance reports.

When the Print Performance Report display appears (Figure 294), type the library name in which you collected the data. Press Enter. The data from the different performance collections is shown.

```

Print Performance Report

Library . . . . . JDE_OW

Type option, press Enter.
  1=System report  2=Component report  3=Transaction report  4=Lock report
  5=Job report    6=Pool report      7=Resource report   8=Batch job trace report

Option  Member      Text                               Date      Time
      OW10AM      Performance test                 02/19/99  15:11:11
      FRIDAY02                                  11/06/98  17:00:21
      FRIDAY01                                  11/06/98  16:21:17

Bottom

F3=Exit  F5=Refresh  F11=Work with your spooled output files  F12=Cancel
F15=Sort by member  F16=Sort by text

```

Figure 294. Print Performance Report main menu

Select the required data collection, and print the following reports by selecting the appropriate option value:

- System Report (all jobs)
- Component Report (all jobs)
- Job Summary Report (all jobs)
- Transaction Detail Report (selected jobs only)
- Transition Report (selected few jobs only)

Other reports such as the Resource (Interval) Report may also be useful in specific cases. For example, the Resource Report can be used to assess line utilization during the different Performance Monitor time intervals.

The System Report provides a good overview of overall system resource utilization.

Figure 295 shows a page out of the Component Report. This page is part of the “Job Workload Activity” and includes Client Access Express database server jobs QZDASOINIT/QUSER/nnnnn. Job 203328 is highlighted because it did significant work as indicated by its CPU utilization and disk I/O (Sync, Async, and Logical) counts.

Component Report													2/19/99 16:48:4				
Job Workload Activity													Page				
7:35 AM test																	
Member . . . : OW10AM Model/Serial : 650-2243/10-1F45M Main storage . . : 12.0 G Started . . : 12/07/98 07:30:1																	
Library . . : JDE_OW System name . S101F45M Version/Release : 4/ 3.0 Stopped . . . : 12/07/98 17:26:2																	
Job Name	User Name	Job Number	T	y	p	t	Pl	Pl	Util	Tns	Rsp	Sync	Async	Logical	I/O	Perm	Write
QZDASOINIT	QUSER	202956	C	02	20	.00	0	0	.000	0	0	0	0	0	0	0	0
QZDASOINIT	QUSER	203328	C	02	20	.052	0	0	.000	220	750	19	0	1	8	81	
QZDASOINIT	QUSER	203417	C	02	20	.000	0	0	.000	0	0	0	0	0	0	0	
QZDASOINIT	QUSER	203418	C	02	20	.000	0	0	.000	0	0	0	0	0	0	0	
QZDASRVSD	QUSER	202958	B	02	20	.000	0	0	.000	0	4	0	0	0	0	0	
QZDSTART	QSNADS	202734	A	02	40	.000	0	0	.000	0	0	0	0	0	0	0	

Figure 295. Component report example

Logical I/Os are the I/Os issued specifically by the AS/400 program. They are called logical because they actually read from or write to data already in main storage as well as cause a physical I/O to a disk device. In general, running the same application with the same number of record/rows repetitively should result in the same number of logical I/O counts each run time.

Physical I/Os mean data was actually read from a disk device or written to the disk device. Physical I/Os may be *synchronous* or *asynchronous*.

High synchronous disk I/O counts can be an indicator of poor performance because the application has to wait for the synchronous disk I/O to complete before continuing processing. High synchronous disk I/Os can be an indication of poor application coding or poor use of indexes or table scans by SQL. Significantly reducing synchronous disk I/Os may often improve performance. High asynchronous disk I/O counts normally are not responsible for poor performance because they do not usually cause the job to wait for the disk I/O completion.

Remember that the Component Report shows the number of disk I/Os over the time the Performance Monitor was collecting data. Therefore, a large disk I/O count may not indicate a disk bottleneck if the monitor was active for 30 minutes or more. The Transaction Report (based on trace data) for a specific job shows the disk I/Os per transaction, which can be a more accurate indication of synchronous disk I/O impact on performance

Possible ways you can reduce synchronous disk I/Os are by using the Set Object Access (SETOBJACC) command, using blocking in the application, and by optimizing the SQL queries.

You see another QZDASOINIT job, job 202956, which did very little work. In an actual customer environment, you may see many QZDASOINIT jobs doing significant work and others doing almost no work. Because subsystem QSERVER is defined to support prestarted jobs for database serving, you see all of the currently active QZDASOINIT jobs on the component report, even if some of them are just waiting for a remote client to perform ODBC functions with the AS/400 system.

When doing capacity planning with BEST/1, you need to ensure that you include only the database server jobs that actually did significant work in any workload definitions.

Note also, that none of the QZDASOINIT jobs show any transaction (“Tns”) count. This is because the system does not identify transactions for non-interactive jobs in the Performance Monitor summary level data, which is used by the Component Report.

The Transaction Report *SUMMARY option uses Performance Monitor *trace* data. Typically, you use the transaction report information when system-wide tuning indicates a performance problem still exists, or is localized to one or two application types running on the system.

The following example Transaction Report displays are shown after selecting option 3, Transaction Report, on the Print Performance Report menu.


```

Print Transaction Report (PRITNSRPT)

Type choices, press Enter.

Member . . . . . >OW10AM           Name
Report title . . . . . >'Performance problem analysis'
Report type . . . . . *SUMMARY       *SUMMARY, *INSACT, *TRS
      + for more values
Time period for report:

Starting time . . . . . *FIRST       Time, *FIRST
Ending time . . . . . *LAST         Time, *LAST

Additional Parameters

Library . . . . . > JDE_OW           Name

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 296. Print Transaction Report display

The Job Summary provides reports on:

- Job level summary giving transaction counts (interactive only; with QZDASOINIT being a server job, it shows 0 transaction counts); average, maximum and minimum resource consumption; and seize/lock conflict counts.
- Top ten information on interactive programs, resource consumption, and response time.
- Seize and lock conflicts.

Figure 297 on page 614 shows an example of the Transaction Report Job Summary section that includes the QZDASOINIT job (203328) previously shown on the Component Report. Note that job 203328 is considered a BJ (batch pre-started job) by the system. Although it did significant work, the job does not have any transactions (Tot Nbr Tns) recorded.

Job Summary Report															2/19/99 17:05:3								
Job Summary															Page 001								
Member . . . OW10AM Model/Serial . . : 650-2243/10-1F45M Main storage . . . : 12.0 G Started . . :12/07/98 07:35:00																							
Library . . . : JDE_OW System name . . . : S101F45M Version/Release : 4/ 3.0 Stopped :12/07/98 07:40:00																							
Job	User Name/Job	*On/Off*	T	P	P	Tot	Resp	Sec	CPU Sec		Average DIO/Transaction				Number	K/T							
Name	Thread	Number	Pl	p	y	g	Tns	Avg	Max	Util	Avg	Max	DBR	NDBR	Wrt	Sum	Max	Sum	Max	Lck	Size	Sec	
QPADEV0044	SSTAT	203298	05	I	20																		
FINPLA0205	CRBRO	203175	10	B	55					7.4	410.61					401		1658					
QPADEV0109	THARB	203308	05	I	20																		
QPADEV0080	QSYSOPR	203311	05	I	20																		
QPADEV0021	CSTEVEN	203318	05	I	20	N	4	.5	6.3	.02	.12	5			5	45	1	6				1	
QZSOSIGN	QUSER	203323	02	B	20						.06					18		12					
QZDASOINIT	QUSER	203328	02	B	20						4.38					455		1466				2	
SEALMSSVR	QPGMR	203333	02	B	25						3.14						10						
QTGTELNETS	QTCP	203336	02	B	20						9.74					5		18				1	
QPADEV0027	KCONR	203337	05	I	20																		
QPADEV0234	FDEAN	203338	05	I	20	N	32	.3	3.5	.02	.26	2	4		6	43	1	6					
QPADEV0144	JEAHA	203339	05	I	20																		
QPADEV0150	SCELL	203341	05	I	20																		

Figure 297. Transaction Report: Job Summary example

Notice that the “Max” (per transaction) counts are shown for the ODBC database server job 203328. The abnormally high count of 455 synchronous disk I/Os and 1,466 asynchronous disk I/Os indicates further investigation of this job is required. Also notice that there are two seize conflicts for this job. You need to look at the Transaction Report and the Transition Report for job 203328.

Although it is not shown in this example, the Job Summary contains other useful information. We recommend that you review at least one other section, “Longest Seize/Lock Conflict”, to see if any other jobs are holding locks on objects such as the SQL tables/files used by the ODBC jobs. High Seize/Lock times can significantly degrade performance. If you see this, consider inappropriate row/record locking for update by the application or unanticipated use of the table/file by another application or access paths being built or maintained. Using higher levels of database commitment control can also be responsible for high or frequent seize/lock time values.

At this stage, print transaction and transition reports for the job 203328.

Figure 298 shows an example of the Transaction Report, Job Transaction report for QZDASOINIT job 203328. Pages of the actual report were selected and shown in this figure. This Transaction Report identifies communication transactions and associated CPU seconds (CPU Sec Per Tns), physical Disk I/Os per transaction, and transaction response time.

Notice the transaction involving OS/400 module QZDASQL caused a transaction response time of 5.811 seconds. This included a seize wait of 2.066 seconds, a lock wait of 1.303 seconds, and 0.151 seconds of CPU time. The remaining time was spent in disk I/Os and CPU wait.

Transaction Report															2/19/99 14:17:0		
Member . . : OW10AM Model/Serial . . : 650-2243/10-1F45M Main storage . . : 12.0 G Started . . :12/07/98 07:30:24															Page 000		
Library . . : JDE OW System name . . : S101F45M Version/Release : 4/ 3.0 Stopped . . . :12/07/98 13:17:56																	
Job name . . : QZDASOINIT User name . . . : QUSER Job number . . . : 203328 TDE/Pl/Pty/Prg . . : 497D/02/20/NO																	
E CPU ---- Physical I/O Counts ---- **** Transaction Response Time (Sec/Tns)*** -EMPL-																	
x Sec ----- Synchronous ---- Async ***** - Activity Level Time- Inel Long C I Seize																	
c Program Per DB DB NDB NDB Disk **** Short Seize Time Wait u n Hold Key																	
Time	p	Name	Tns	Read	Wrt	Read	Wrt	Sum	I/O	** Active	Wait	Cft	A-I/W-I	Lck/Oth	r l	Time	Thin
07.35.54		QZDASOINIT	.002					0	0	.004	.002					8	.1
07.35.54		QZDASOINIT	.101			13		13	45	1.156	1.155					8	.2
07.36.01		QZDASOINIT	.007					0	0	.008	.001					6	.3
07.36.01		QZDASOINIT	.001					0	0	.005	.004					6	.1
07.36.01		QZDASOINIT	.051	1	3	2	4	10	20	.362	.357					6	.1
07.36.04		QZDASOINIT	.001					0	0	.008	.007					8	2.1
07.36.06		QZDASOINIT	.008					0	0	.012	.004					6	2.3
07.36.06		QZDASOINIT	.004					0	0	.007	.003					6	.4
07.36.06		QZDASOINIT	.004					0	0	.005	.001					6	.1
07.36.16		QZDASOINIT						0	0	1.004	1.004					8	10.1
07.36.18	Y	QZDASQL	.151	1	2	2	12	17	105	5.811	5.506	2.066		1.303	7	6.6	.2
07.36.35		QZBSCOMM	.127				2	2	28	2.296	5.295						.1
07.36.49		QZBSCOMM	.002					0	0	.002	.000					11	9.2

Figure 298. Transaction Report: Job Transaction Report example

The Transition Report for the transaction at 7.36.18 (Figure 299) indicates the objects involving locks and seize conflicts. This report also shows that the OS/400 module QSQCONN (SQL Connection) was in control when high number of I/Os occurred.

Transition Report															2/19/99 16:17:2		
Member . . : OW10AM Model/Serial . . : 650-2243/10-1F45M Main storage . . : 12.0 G Started . . :12/07/98 07:35:53															Page 000		
Library . . : JDE OW System name . . : S101F45M Version/Release : 4/ 3.0 Stopped . . . :12/07/98 07:36:42																	
Job name . . : QZDASOINIT User name . . . : QUSER Job number . . . : 203328 TDE/Pl/Pty/Prg . . : 497D/02/20/NO																	
Job type . . : BJ Elapsed Time- Sec Sync/Async Phy I/O -MPL-																	
State Wait Long Active Inel CPU DB DB NDB NDB C I Last 4 Programs in Invocation Stack																	
Time	W A I Code	Wait	Sec	Read	Wrt	Read	Wrt	Tot	r l	Last	Second	Third	Fourth				
07.36.16.809	->A	10.085															8
07.36.16.810	W																8
07.36.18.814	W<-		1.004														8
	QZDASOINIT		1.004*			0	0	0	0	0*							7
07.36.18.946	->A	.132	.001	.001													7
07.36.19.856	SZWT	*/	.913/*														
07.36.25.538	LKRL																
07.36.26.588	SZWT	*/	1.153/*														
07.36.27.742	LKWT	*/	1.303/*														
07.36.27.742	W<-		2.796	.030		1	2	4	9	QSPCLOSE	QDMCLOSE	QMHLJLOG	QMHSRUL				
07.36.29.045	->A	1.303	.001	.001		3		1	9	PAG=	0		FWrt=	9			
07.36.31.311	LKRL																
07.36.35.354	W<-		1.310	.119	1	1	6	8	9	QSQCONN	QSQXCUTE	QZDASQL	QZDAMDPA				
								95		PAG=	0	FWrt=	109				
											Dec =	0	Bin =	1	Flp =	0	
	QZDASQL		5.811*	.151	1	2	2	12	17*	PAG=	0	FWrt=	118				
					0	9	0	96	105		Dec =	0	Bin =	1	Flp =	0	
07.36.35.356	->A	.001	.001	.001					9								
07.36.40.598	SZRL																
07.36.40.651	W<-		5.295	.126			2		9	QWTCGET	QZBSCOMM	QZBSCOMM	QZDASOIN2				
							28			PAG=	0	FWrt=	30				
	QZBSCOMM		5.296*	.127	0	0	0	2	2*	PAG=	0	FWrt=	30				
					0	0	0	28	28								

Figure 299. Transaction Report: Job Transaction Transition example

For further details on interpretation of these reports, refer to Chapter 8 in *AS/400 Client/Server Performance Using Windows Clients*, SG24-4526.

21.3.5 AS/400 database query optimization tools

The DB2 UDB for AS/400 database has a sophisticated methodology used in optimization. With this in mind, certain tools are essential when dealing with analyzing the optimizer. The tools are needed to know what the queries and SQL you write are doing, or what the optimizer is doing with them. You need the following tools to analyze your queries and the optimizer:

- Database monitor statistics
- Print SQL information
- Query optimizer debug messages
- Change query attributes

21.3.5.1 Database Monitor statistics

Database Monitor statistics log all information and statistics about queries being run on a particular job or at a system level. The STRDBMON command is used to start the monitor, and the ENDDDBMON command is used to end it. Database Monitor can also be started from the STRPFRMON command by using the DBMON parameter.

The monitor can be specified for a specific job or all the jobs on the system, with statistics placed in the output database file specified on the command. Each job in the system can be monitored concurrently by two monitors: one monitor for the specific job and one for all jobs on the system. When monitored by two monitors and each monitor is logging to a different output file, monitor records are written to both files. Records in the output database files are uniquely identified by their record identification number.

Some of the database information collected includes:

- System and job name
- SQL statement and sub-select number
- Start and end timestamp
- Estimated processing time
- Total rows in file queried
- Number of rows selected
- Estimated number of rows selected
- Estimated number of joined rows
- Key fields for advised index
- Total optimization time
- Join type and method
- ODP implementation

Running the Database Monitor allows you to answer several key questions relative to your database, such as:

- How many queries perform dynamic replans?
- How many temporary indexes were created over a particular file?
- Which queries are the most time consuming?
- Which user is running those queries?
- Which queries were implemented using reusable ODPs?
- Does a particular PTF apply to my users?
- Has the implementation of a particular query changed with the application of a PTF or a new release?

The questions are actually answered by querying the information collected from running the Database Monitor.

The Database Monitor data is collected in 14 different records (13 records if *SUMMARY data was collected). For V4R3, Section 1.4.7, “Monitoring Database Query Performance” in *DB2 for AS/400 Database Programming*, SC41-5701, gives you a detailed description of each record type and DDS source for logical files based on the data collection physical file. For V4R4, go to the AS/400 Information Center at:

<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>

At this site, select **Database and File Systems->Database Management->DB2 Universal Database for AS/400 books online**.

Open the SQL Programming book and find the “Database Monitor Statistics” section of Chapter 26, “Monitoring and Optimizing Query Performance Tools”.

Before starting to analyze these records, you may have to create individual logical files for each record type using the supplied logical file DDS source. Create these source files in a library of your choice (for example, DBMONLF). Then, build logical files for the collected Database Monitor data by using these source files.

The following section contains sample CL and Command source programs to help you build the logical files.

The CRTDBMONF command

```

CMD          PROMPT('Create DBMON Logical Files')
PARM        KWD(CUSTLIB) TYPE(*NAME) MIN(1) +
           PROMPT('Customer performance data lib')

```

CL source CRTDBMONFP program

```

PGM          PARM(&CUST_LIB)
DCL          VAR(&CUST_LIB) TYPE(*CHAR) LEN(10)
MONMSG      MSGID(CPF0000)
DLTF        FILE(&CUST_LIB/QQQ1000)
DLTF        FILE(&CUST_LIB/QQQ3000)
DLTF        FILE(&CUST_LIB/QQQ3001)
DLTF        FILE(&CUST_LIB/QQQ3002)
DLTF        FILE(&CUST_LIB/QQQ3003)
DLTF        FILE(&CUST_LIB/QQQ3004)
DLTF        FILE(&CUST_LIB/QQQ3005)
DLTF        FILE(&CUST_LIB/QQQ3006)
DLTF        FILE(&CUST_LIB/QQQ3007)
DLTF        FILE(&CUST_LIB/QQQ3008)
DLTF        FILE(&CUST_LIB/QQQ3010)
DLTF        FILE(&CUST_LIB/QQQ3014)
DLTF        FILE(&CUST_LIB/QQQ3018)
DLTF        FILE(&CUST_LIB/QQQ3019)
CRTLF       FILE(&CUST_LIB/QQQ1000) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3000) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3001) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3002) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3003) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3004) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3005) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3006) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3007) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3008) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3010) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3014) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3018) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)
CRTLF       FILE(&CUST_LIB/QQQ3019) SRCFILE(DBMONLF/QDDSSRC) MAINT(*IMMED)

```

Once you create the logical files, you can start the database and SQL query-related analysis. A number of Database Monitor examples are provided in

Section 1.4.7.5 of the *DB2 for AS/400 Database Programming*, SC41-5701, or in Chapter 26 of the *V4R4 SQL Programming* book in the AS/400 Information Center. You may use these examples or build your own queries or reports.

21.3.5.2 Useful tips for Database Monitor analysis

The Database Monitor collects a lot of data in too many records and fields. For simplicity, this section provides information on the most often used record types and fields, as well as a set of key database analysis example queries. Most of these example queries do not even require creation of logical files.

Most commonly used record types

The most commonly used Database Monitor records types are:

1000	SQL summary record
3000	Arrival sequence
3001	Using existing index
3002	Index created
3003	Query sort
3006	Access path rebuilt
3007	Index optimization data
3010	Host variable and ODP implementation

The fields QQJOB (Job name), QQUSER (Job user name), QQJNUM (Job number), and QQTIME (Time that the record was created) are common to all these record types.

Commonly used fields and performance analysis examples

Other most commonly used fields for each record type are shown here, along with performance analysis examples:

- **1000 Record Type:** SQL statement summary

This record type contains basic SQL query information for each SQL query operation (open, update, close, commit and so on). The commonly used fields are:

QQ1000	Prepared text for SQL statement
QQC21	Type of SQL operation (OP, FE, CL, UP, IN, DL ...)
QQI2	Number of rows fetched (only on FE records)
QQI3	Number of rows updated/ inserted/ deleted
QQI5	Refresh counter
QQI4	Elapsed time for this operation in <i>milliseconds</i>
QQI6	Elapsed time for this operation in <i>microseconds</i>
QQSTIM	Timestamp for start of SQL operations
QQETIM	Timestamp for end of SQL operation

If you need to identify jobs which need further analysis, you can run the following query on this record type:

```
SELECT SUM(QQI4), COUNT(*), QQJOB, QQUSER, QQJNUM
FROM QAPMDBMON
WHERE QQRID = 1000
GROUP BY QQJOB, QQUSER, QQJNUM ORDER BY 1 DESC
```

To find which SQL operations account for the most run time for selected jobs, you may run this query:

```
SELECT SUM(QQI4), COUNT(*), QQC21
FROM QAPMDBMON
```

```
WHERE QQRID = 1000 AND QQJNUM = 'job_number'
GROUP BY QQC21 ORDER BY 1 DESC
```

• **3000 Record Type: Arrival Sequence (Table Scan)**

This record type points out queries where an entire table is scanned without using an index. This is generally OK in cases where a large portion of the file is selected or if file contains a small number of records.

- QQPTFN** File name
- QQPTLN** File library name
- QQTOTR** Number of rows in table (use to determine if table scan is for a significant number of rows)
- QQRCOD** Reason code (why arrival sequence chose)
- QQIDXA** Index is advised (Y or N) if “N”, QQI2, and QQIDXD do not contain data; index is not advised if an index for these keys already exists
- QQI2** Number of primary keys in QQIDXD field
- QQIDXD** Suggested keys for index (selection only)

For example, by running the following query, you can list the advised indexes for all those SQL queries that performed a table scan (selection only):

```
SELECT QQTLN, QQTFN, QQIDXD
FROM QAPMDBMON WHERE QQRID = 3000
AND QQIDXA = 'Y'
```

To determine which queries involve arrival sequence, you can use the following code:

```
SELECT SUM(QQI4), COUNT(*), QQUCNT,QQ1000
FROM QAPMDBMON WHERE QQJNUM =
'job_number'
AND QQRID = 1000 AND QQUCNT IN
(SELECT DISTINCT QQUCNT FROM QAPMDBMON
WHERE QQRID = 3000 AND QQJNUM = 'job_number')
GROUP BY QQUCNT, QQ1000 ORDER BY 1 DESC
```

• **3001 Record Type: Using Existing Index**

This record type shows the file and index that are used to access the file, and why it was chosen. If the index was chosen for join, additional information is given to help determine how file “fits” in the join. The order of 3001 records indicates join order chosen by optimizer:

- QQPTFN** File name (*N indicates temporary results table)
- QQPTLN** File library name (blank if file name is *N)
- QQIFNM** Name of selected index (may be *TEMP)
- QQILNM** Index library name (blank if index name is *TEMP)
- QQRCOD** Reason the index was selected
- QQC21** Join method (NL, MF, HJ)
- QQC22** Join type (IN, PO, EX)
- QQC23** Join operator (EQ, LT, GT, CP and so on)
- QQTOTR** Number of rows in table
- QQAJN** Estimated number of joined rows from this table
- QQIDXA** Index advised (Y or N)

QQI2 Number of suggested primary keys in QQIDXD field
QQIDXD Recommended primary and secondary keys for index on this table

• **3002 Record Type: Index Create**

This record type contains data on the temporary indexes created for running a query. The reason for temporary index creation is placed in the record type 3007:

QQPTFN Table name for which index is built, Name of *N indicates temporary results table (3004 record) - index build unavoidable

QQPTLN Table library name (blank if file name is *N)

QQIFNM Name of index. Will be blank if index built over entire table contains the index name if an index from index build occurred

QQILNM Index library name (blank if build on entire table)

QQRCOD Reason the index build was done

QQTOTR Number of rows in table

QQRIDX Number of entries in temporary index

QQ1000 Contains join, order by, or group by keys used in index build

QQSTIM Timestamp for start of index build

QQETIM Timestamp for end of index build

QQSTIM QQSTIM shows elapsed time for index build

QQI2 Number of primary selection keys in QQIDXD

QQIDXD Suggested primary and secondary keys for index

For example, you may use the following query to determine which indexes are built most often:

```
SELECT QQUCNT, QQETIM-QQSTIM, QQPTFN, QQTOTR, QQRIDX, QQRCOD,
QQIDXA, SUBSTR(QQ1000, 1, 100), SUBSTR(QQIDXD, 1, 200)
FROM QAPMDBMON
WHERE QQRID = 3002 AND QQIDXA = 'Y'
ORDER BY 2 DESC
```

• **3003 Record Type: Query Sort**

QQSTIM Timestamp for start of refill and sort

QQETIM Timestamp for end of refill and sort

QQRCOD Reason for choosing query sort

QQRSS Number of rows in sort space

• **3006 Record Type: Access Plan Rebuild**

QQRCOD field lists the reason the replan is occurring

• **3007 Record: Index Optimization and Selection**

This record type shows all indexes that are evaluated for a given file, including which one (if any) was selected for use in this query and which were not (and why).

QQPTFN File name

QQPTLN File library name

QQC11 Optimizer timed out (Y or N)
QQ1000 Contains library qualified index names, each with a reason code

• **3010 Record Type:** Host Variable and ODP Implementation

This record type shows the substitution values for the host variables or parameter markers in the query text:

QQ1000 Host variable or parameter marker values
QQI5 Refresh count (use to link to specific query instance)

Note on Index Advisor

Query optimizer analyzes the record selection in the query and determines, based on the default estimate, if creation of a permanent index can improve performance. If a permanent index can be beneficial, it returns the key fields necessary to create the suggested index. The index advisor information can be found in debug message CPI432F or in the Database Monitor records QQQ3000, QQQ3001, and QQQ3002. The main point here is to know that the intention of creating the suggested index is to improve the performance of the query.

21.3.5.3 Print SQL Information

Print SQL Information allows you to print the SQL information contained within a program, SQL package, or service program. A spooled file is created that contains SQL statements, the type of access plan used by each statement, and such commands as CRTSQLxxx used to invoke the SQL precompiler. The output information is similar to the debug messages.

When a client/server application runs, it selects a data source that contains the location of the AS/400 database. The data source is defined in the ODBC.INI file. The SQL package name used by the application is described in the PACKAGE specification. The AS/400 object type for SQL packages is *SQLPKG.

Use the PRTSQLINF command with object type *SQLPKG to print the package.

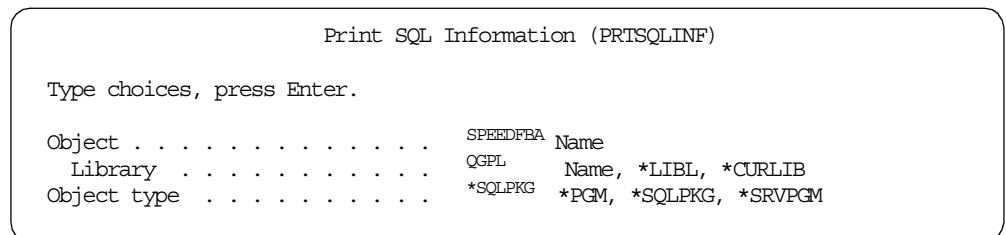


Figure 300. Print SQL Information display

Checking the description of the SQL Package indicates when the package was created. In a production client/server application, the creation date of the package should reflect when the application went “live”.

Figure 301 on page 622 shows an example of PRTSQLINF report. The time stamp against the access plan represents when the plan was last updated. In a stable environment, this should correspond to the time the application was first run. In an environment where a significant number of records or rows are added

or deleted, or index fields are updated, the time stamp changes when the file is next opened if the Query Optimizer determines the access plan is no longer efficient.

The access method identified in the access plan may help to identify the need to create new logical views, for example, to improve performance. If the access method chosen at program creation time is already known to not be appropriate, you can create the appropriate index or logical view before you go into production mode.

Remember, when using AS/400 Extended Dynamic SQL, the order of the SQL statements within the package are the order in which they were received by the AS/400 Query Optimizer. The time stamps represent the last time the access plan for the same statement was updated. This update occurs during file/table open processing.

```
5769SS1 V4R3M0 980729 Print SQL information SQL package QGPL/OEXPLORFJA 02/17/99 15:47:24 Page 1
Object name.....QGPL/OEXPLORFJA
Object type.....*SQLPKG
 CRTSQL***
  PGM(QGPL/OEXPLORFJA)
  SRCFILE( / )
  SRCMBR( )
  COMMIT(*NONE)
  OPTION(*SYS *PERIOD)
  TGIRLS(*PRV)
  ALWCPYDTA(*OPTIMIZE)
  CLOSQLCSR(*ENDPGM)
STATEMENT NAME: QZ7E66F4FE50FB0000
SELECT * FROM SYSB733/F00941 WHERE ( LMLL = ? )
SQL4021 Access plan last saved on 02/10/99 at 09:28:27.
SQL4020 Estimated query run time is 1 seconds.
SQL4027 Access plan was saved with DB2 Symmetric Multiprocessing installed on the system.
SQL4017 Host variables implemented as reusable ODP.
SQL4006 All access paths considered for file 1.
SQL4008 Access path F00941_1 used for file 1.
SQL4011 Key row positioning used on file 1.
STATEMENT NAME: QZ7E66F4FE7F0F8000
SELECT * FROM SYSB733/F00942 WHERE ( EMPATHCD = ? )
SQL4021 Access plan last saved on 02/10/99 at 09:28:28.
SQL4020 Estimated query run time is 1 seconds.
SQL4027 Access plan was saved with DB2 Symmetric Multiprocessing installed on the system.
SQL4017 Host variables implemented as reusable ODP.
SQL4008 Access path F00942_1 used for file 1.
SQL4011 Key row positioning used on file 1.
STATEMENT NAME: QZ7E66F4FE8D448000
SELECT * FROM SYSB733/F00945 WHERE ( RMRLS = ? )
SQL4021 Access plan last saved on 02/10/99 at 09:28:28.
SQL4020 Estimated query run time is 1 seconds.
SQL4027 Access plan was saved with DB2 Symmetric Multiprocessing installed on the system.
SQL4017 Host variables implemented as reusable ODP.
SQL4008 Access path F00945_1 used for file 1.
SQL4011 Key row positioning used on file 1.
```

Figure 301. Print SQL Information example

21.3.5.4 Debug messages

Debug messages issue informational messages to the job log about the implementation of a query. The messages describe query implementation

methods such as indexes, file join orders, temporary results, access plans, and ODPs (Open Data Paths).

The messages explain what happened during the query optimization. Questions, such as why an index was used or not used, or why a temporary result was required, are explained. It also shows the join order of the files and the index advised by the optimizer.

One of the most significant steps that can be used when solving any programming problems, is to isolate problem code into smaller units. For example, if you identified a performance problem with an SQL query (such as through the use of STRDBMON), you may copy the SQL statement (and the required host variable definitions) to test environment and debug the isolated SQL query.

Note

It is often difficult to determine which server job corresponds to a particular client. You can write an exit program that can be used with QIBM_QZDA_INIT to start and stop debug for all the new jobs. Also, remember that if you turned the job log to *NOLIST, you will not see any debug messages.

Run the selected SQL statement in the debug mode by using the following command:

```
STRDBG UPDPROD(*YES)
```

By running this command, most of the SQL optimizer messages are displayed in the job log. The query optimizer provides you with information messages on the current query processing. Display the job log by using the DSPJOBLOG command to see the optimizer messages. Figure 302 shows the resulting display.

```
Display All Messages
System: ASM23
Job . . . : QPADEV0131 User . . . : owuser Number . . . : 610836

Access path built for file QCUSTCDT.
ODP created.
Blocking used for query.
ODP deleted.
Access path built for file QCUSTCDT.
ODP created.
Blocking used for query.
ODP deleted.
Access path built for file QCUSTCDT.
ODP created.
Blocking used for query.
Connection to relational database RCHASBDS ended.
SQL cursors closed.

Press Enter to continue.

F3=Exit F5=Refresh F12=Cancel F17=Top F18=Bottom
```

Figure 302. First level text of job log messages

If you need more information on what the optimizer did, for example, why the access path was built, analyze the messages using the Help key for the second level text. If you positioned on the first message on the display shown in Figure 302, the display shown in Figure 303 may appear.

```

Additional Message Information

Message ID . . . . . : CPI4321      Severity . . . . . : 00
Message type . . . . . : Information
Date sent . . . . . : 02/23/99      Time sent . . . . . : 09:45:10

Message . . . . . : Access path built for file QCUSTCDT.
Cause . . . . . : A temporary access path was built to access records from
member CUSTCDT of file QCUSTCDT in library QIWS for reason code 1. This
process took 0 minutes and .0 seconds. The access path built contains 13
entries. The access path was built using 0 parallel tasks. A zero for the
number of parallel tasks indicates that parallelism was not used. The reason
codes and their meanings follow:
  1 - Perform specified ordering/grouping criteria.
  2 - Perform specified join criteria.
  3 - Perform specified record selection to minimize I/O wait time.
The access path was built using the following key fields. The key fields
and their corresponding sequence (ASCEND or DESCEND) will be shown:
More...

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details  F12=Cancel
F21=Select assistance level

```

Figure 303. Second level text of job log messages

You can evaluate the structure and performance of the given SQL statement in your programs using the informational messages entered in the job log. For a complete list of optimizer messages and their interpretation, refer to Section 1.4.4 in *DB2 for AS/400 Database Programming*, SC41-5701.

Predictive query governor tip

Before running the SQL statement, you may set the time limit of ZERO for the parameter QRYTIMLMT in the CHGQRA command. The query optimizer messages are written to the job log prior to the actual execution of the query. This way, you can get all the optimizer messages without actually running the query.

To get detailed job information for a prestarted job QZDASOINIT, you may want to trace the job by using Job Trace facility of AS/400 (see to 21.3.8.1, “Job trace” on page 651).

21.3.5.5 Query attributes

A user can change attributes of a query during the execution of a job. Some of the attributes that can be modified are the query processing time limit, parallel processing degree, and the number of tasks.

The types of attributes that can be modified are:

- Predictive query governor
- Query parallelism

- Asynchronous job
- Apply CHGQRYA to remote

Important points to remember are that the query time limit is checked against the estimated elapsed query time before initiating a query. The inquiry message CPA4259 is displayed showing the predicted runtime and what operations the query will perform. Time limit of zero is used to see the optimization messages without actually running the queries. Debug messages are written to the job log if the query is canceled.

21.3.5.6 Comparison of optimization analysis methods

Table 48 compares the features and optimization tools discussed in the previous sections.

Table 48. Comparison of optimization analysis methods

PRTSQLINF	STRDBG/CHGQRYA	STRDBMON
Available without running query (after access plan is created).	Only available when the query is run.	Only available when the query is run.
Displayed for all queries in SQL program, regardless of whether it is executed.	Displayed only for those queries that are executed.	Displayed only for those queries that are executed.
Information on host variable implementation.	Limited information on the implementation of host variables.	All information on host variables, implementation, and values.
Available only to SQL users with programs, packages, or service programs.	Available to all query users (OPNQRYF, QUERY/400, SQL).	Available to all query users (OPNQRYF, QUERY/400, SQL).
Messages printed to a spooled file.	Messages displayed in job log.	Performance records written to a database file.
Easier to tie messages to a query with subqueries or unions.	Difficult to tie messages to query with subqueries or unions.	Uniquely identifies every query.

21.3.6 Database performance analysis: Operations Navigator

The *Database* component of Operations Navigator provides a graphical interface for many DB2 Universal Database (UDB) for AS/400 database operations, including:

- Creating and managing tables and views
- Creating and managing OS/400 journals (record changes to database and other functions supporting journals)
- Entering new or modifying already created SQL statements
- Running and debugging previously created SQL statements (referred to as *scripts*)
- Saving SQL statements for later use
- Doing performance analysis of your SQL statements

OS/400 V4R5 brought the following enhancements to the Database component of Operations Navigator:

- Visual Explain: You can use Visual Explain to graphically display the implementation of your SQL statements that were run either through Run SQL Scripts or in a separate job. You can use this tool to see information about both static and dynamic SQL statements. You can also use this tool to better understand where the highest costs of your queries are taking place. Figure 304 shows an example of a Visual Explain screen.
- SQL Performance Monitor: Detail monitor
- Display current SQL statement for a job (if exists)
- Display table, view, and index description
- Display record locks
- Allow a user to work with system libraries and make a user's list persistent

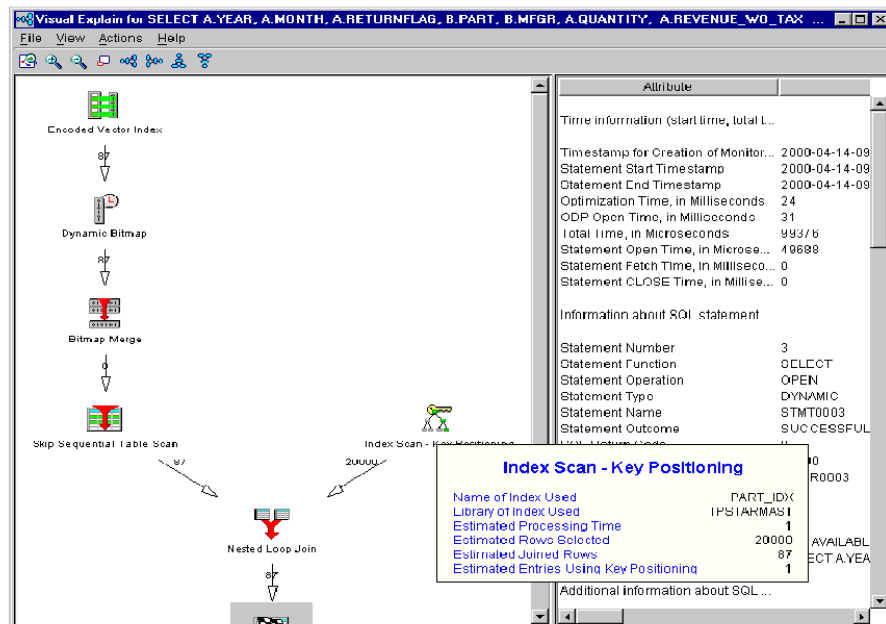


Figure 304. Visual Explain sample display

The Database component of AS/400 Operations Navigator is not installed by default when choosing the *Typical* installation option of IBM AS/400 Client Access Express. If it is not currently installed, you can run Selective Setup to install it.

With proper authorization to the database objects, the user of the database graphical interface has easy access to OS/400 server administration tools, has a clear overview of the entire database system, can perform remote database management, and receives assistance for complex tasks.

For OS/400 V4R4, key enhancements to DB2 UDB for AS/400 include an interface to the SQL-specific Performance Monitor, and new universal database Object Relational Support functions, such as various types of binary large objects (LOBs), User Defined Data Types (UDTs), User Defined Functions (UDFs), and DataLinks.

OS/400's integrated DB2 UDB for AS/400 support is one of the major strengths of AS/400 systems. A complete description of this support is beyond the scope of this redbook. Good sources for details of DB2 UDB for AS/400 capabilities include:

- AS/400 Information Center (<http://www.as400.ibm.com/infocenter>). Once you reach this site, select **Database and File Systems->Database management**. Under Database management, select **DB2 Universal Database for AS/400 books online** to find a list of publications that contain even more information. In this redbook, we list most of these publications here.
 - *Database Programming*, SC41-5701. This book describes database capabilities, primarily outside of SQL terminology. This includes physical files (correspond to SQL tables), logical files (correspond to SQL views), fields (correspond to SQL columns), records (correspond to SQL rows), file management and file security
 - *DB2 UDB for AS/400 SQL Programming*, SC41-5611
 - *DB2 UDB for AS/400 SQL Reference*, SC41-5612
 - *Distributed Data Management*, SC41-5307
 - *DB2/400 Advanced Database Function*, SG24-4249
 - *Developing Cross-Platform DB2 Stored Procedures*, SG24-5485
 - *DB2/400: Mastering Data Warehousing Functions*, SG24-5184
 - *DB2 UDB for AS/400 Object Relational Support*, SG24-5409
- Self study lab exercise with a sample OS/400 database, installation instructions, and lab instructions that can be downloaded from PartnerWorld for Developers, AS400 (AS/400 Partners in Development) Web site:
<http://www.as400.ibm.com/developer>

Select **Education->Internet Based Education->Business Intelligence->Piloting DB2 UDB for AS/400 with Operations Navigator**.

Under OS/400, you can use SQL interfaces to access a *database file* or an *SQL table* as these terms refer to the same object, classified within the OS/400 as an *FILE object type. You can use SQL interfaces to access the file regardless of whether the object was created with the OS/400 Create Physical File (CRTPF) command or SQL CREATE TABLE. OS/400 also supports access to the physical file/table through a logical file or an SQL view - Create Logical File (CRTLFL) command or SQL CREATE VIEW).

Table 49 on page 628 shows the corresponding OS/400 term and SQL term for physical files/tables, records/rows, fields/columns, logical files/views, aliases, and indexes.

Table 49. OS/400 term and SQL term cross reference

OS/400 create statement/term	SQL create statement	OS/400 object type	OS/400 object attribute	SQL term
CRTPF	CREATE TABLE	*FILE	Physical File (PF)	Table
CRTL	CREATE VIEW	*FILE	Logical File (LF)	View
CRTDDMF	CREATE ALIAS	*FILE	DDM File (DDMF)	Alias
CRTI	CREATE INDEX	*FILE	Logical File (LF)	Index
Field				Column
Record				Row

Note: A DDMF represents a Distributed Data Management File. This is the original OS/400 object used to link an OS/400 program's file open to a file on a remote system. In the context of Table 49, an alias created by SQL has no remote system specification. To determine if the DDMF/alias has any remote system specification, you can use the Work with DDM File (*WRKDDMF*) command.

Throughout the remainder of this section, we use the SQL terms “table”, “row”, and “column”, more frequently than their corresponding OS/400 terms “file”, “record”, and “field”. In some cases, we use both corresponding terms, such as “field/column”.

Note: OS/400 supports an object type of table (*TBL). This object type is for data translation.

21.3.6.1 DB2 UDB for AS/400 through Operations Navigator overview

Click the + (plus) sign next to the Database function for the system to which you are attached to see the three major function areas shown in the left pane and right pane in Figure 305.

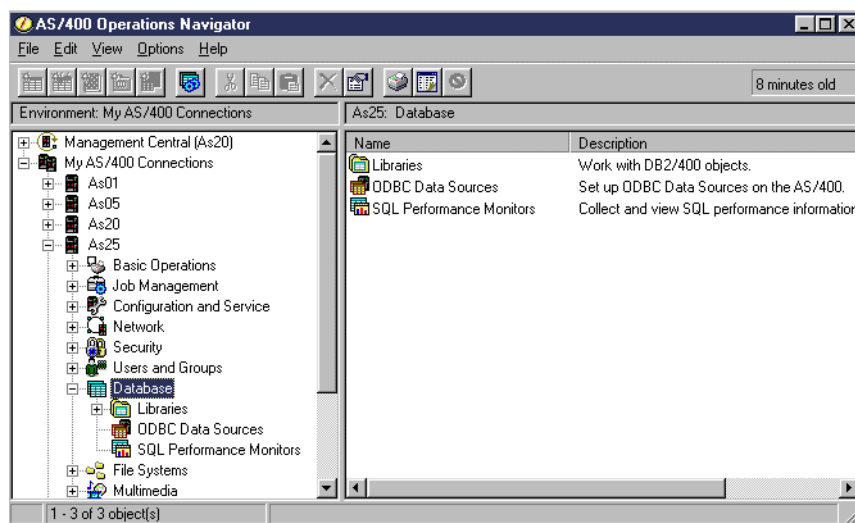


Figure 305. Database function functional areas

There are several other ways to get the same three database function areas to also appear in the right pane as shown. We discuss some of those ways in this

section. However, Operations Navigator Database capabilities are actually grouped under *four* functional branches:

- Database
- Libraries
- ODBC Data Sources
- SQL Performance Monitors

The following sections summarize the capabilities under each of these four major database function groupings. Examples and tips on usage are given for selected sub-functions under each major function group to highlight Operations Navigator interfaces into the wide range of DB2 UDB for AS/400 capabilities.

We do not explain every action on every pull-down menu in this section. We explain the actions that are most significant. Actions such as Explore, Open, Shortcuts, and Print options are similar to these same actions for Operations Navigator in general. For some other database-specific actions or options, you must refer to the Operations Navigator online help information.

For the database functions described in the following sections, you need the appropriate authority to perform the functions.

You can use the SQL GRANT and REVOKE statements to define authority to a table, view, procedure, user-defined functions, and user-defined types. For tables and views, these statements may also specify “processing authority”, such as SELECT (read), INSERT (write), DELETE, and UPDATE.

SQL GRANT and REVOKE can also specify column level authorities.

The Operations Navigator *Database* interface supports table, view, procedure, column, and so forth levels of authority through the *Permissions* action by right-clicking on the database object name within a library.

An alternative to column level authority is to use an SQL CREATE VIEW to a table or a Create Logical File (CRTLF) command based on a file and specify only certain columns/fields. Then you specify authorities/permissions to the logical file or view.

SQL CREATE VIEW or CRTLF can also specify compare values for columns/fields that limit the rows/records that can be seen by those authorized to the view/logical file.

For additional authority details on the Object Relational Support items (functions and types), refer to *DB2 UDB for Object Relational Support*, SG24-5409.

AS/400 SQL software requirements

Base OS/400 provides SQL “runtime support”, not “program development for SQL support”. Runtime support includes the following uses of SQL with no SQL software installation required:

- All Open Database Connectivity (ODBC) support, which includes Operations Navigator functions and Run SQL Scripts jobs and client workstation jobs using Client Access ODBC support such as a Visual Basic program
- All Java Database Connectivity (JDBC) support, which includes client workstation Java applets and local AS/400 Java servlets accessing JDBC
- DB2 UDB for AS/400 support from an already compiled (created) local AS/400 program using embedded SQL in the RPG, COBOL, or C program
- DB2 UDB for AS/400 support from an already compiled (created) local AS/400 program using the SQL CLI (Call Level Interface) in RPG, COBOL, C, or Java
- DB2 UDB for AS/400 support from an interactive 5250 workstation using the Start SQL Session (STRSQL) command

Using DB2 Query Manager support or compiling (create) local AS/400 programs using embedded SQL, such as AS/400 RPG, COBOL and C programs, requires licensed program DB2 Query Manager and SQL Development Kit for AS/400, 5769-ST1. This is for program development support.

21.3.6.2 ODBC Data Sources overview

Open Database Connectivity (ODBC) is a standard interface for database connectivity defined by the Microsoft Corporation. ODBC establishes the standard interface to any database as Structured Query Language (SQL).

In general, the ODBC architecture accounts for an application using the ODBC interface, an ODBC Driver Manager, one or more ODBC Drivers, and an ODBC Data Source (place where the data is stored).

Client Access Express for AS/400 provides the AS/400 ODBC Driver that runs on the client workstation and the ODBC Data Source support that runs on the AS/400 Database Server (production mode job name starts with QZDASOINIT).

With ODBC Data Sources, you can set up a Client Access Express ODBC Data Source by providing a data source name (a name meaningful to you) and an AS/400 system name. An ODBC Data Source consists of the data that the user wants to access and its associated operating system, Database Management System (DBMS), and network platform (if any) used to access the DBMS.

Setup information is associated with a data source, and may include, for example, data formatting and performance options. Data formatting options include qualified name separators, date and time formats, and data translation. Performance options include when to use record blocking, data compression, or an SQL Package. An SQL package stores previously parsed SQL statements to improve performance when used later.

You can also specify if Secure Socket Layer (SSL) is to be used with the ODBC connection. Since SSL may reduce performance significantly, we recommend that you use SSL only if you need it.

Some client applications (including Operations Navigator) may provide their own unique data source definition.

A good source for more information on AS/400 ODBC support is *AS/400 Client Access Express ODBC User's Guide*, SC41-5509.

21.3.6.3 IBM-provided ODBC Data Sources

With Operations Navigator, you can create your own data source to limit the libraries that can be used and, as previously described, your own set of name separators, date and time formats, performance options, and so on.

OS/400 provides two data sources that you should understand even if you are not creating your own data source:

- The data source used by Operations Navigator itself to perform its functions. This data source is identified by the system name you are connected to. For example, if your system name is As25, the data source used by Operations Navigator is named QSDN_As25.

Note: Unless you are an ODBC expert, do not change any of the default settings for this data source. If you change them, Operations Navigator may fail to operate correctly.

- The data source used if you select Database->Run SQL Scripts. The first time you select the action to Run SQL Scripts to a specific AS/400 system, OS/400 creates a data source named, in our example, QDS2_As25. You do not have to create your own ODBC Data Source and understand the data source parameters to run SQL statements against libraries and files or tables to which you are authorized.

21.3.6.4 ODBC Data Source set up parameters

To create your own ODBC Data Source, right-click **ODBC Data Sources** (refer to Figure 305 on page 628), and select **New Data Source**. You are presented with a set of panels where you can name the ODBC Data Source, the name of the associated system, and all the set up parameters.

This section shows you ODBC Data Source setup parameters used by Operations Navigator for the Run SQL Scripts function. You can create your own ODBC Data Source using these parameters as a reference.

To display or modify ODBC Data Source QDS2_system-name, right-click **Database**, and select **Run SQL Scripts**. From the Run SQL Scripts menu bar, select **Connection->ODBC Setup...** to access a panel similar to the one shown in Figure 306 on page 632.

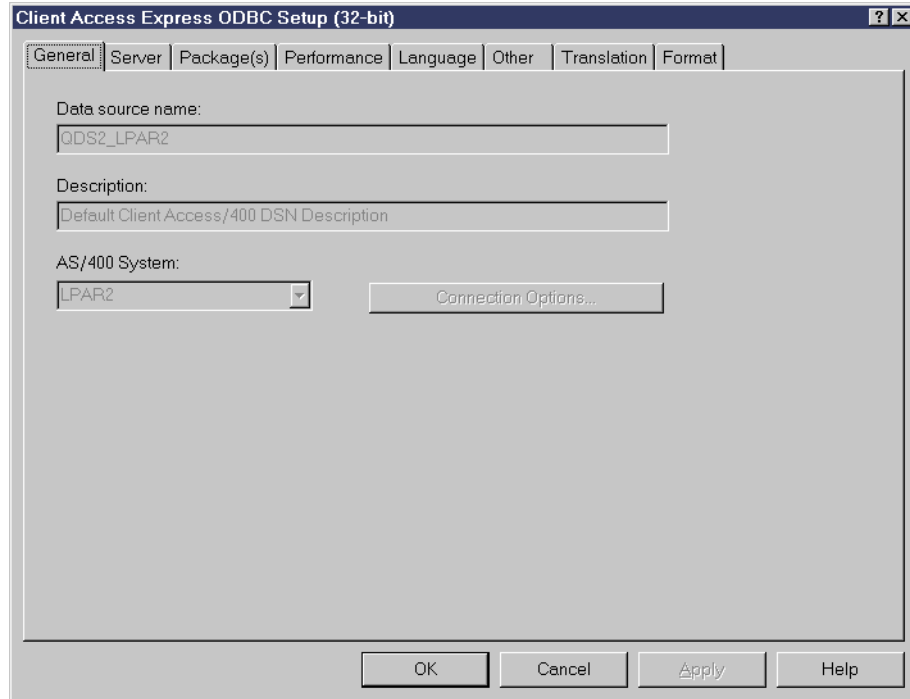


Figure 306. ODBC Data Source: Run SQL Scripts General tab

Here, you see the IBM provided ODBC Data Source name and description and the attached system name. These values (QDS2_AS25, ... AS25 in this example) are grayed out because this system-supplied ODBC Data Source for Run SQL Scripts name cannot be modified. We do not show you all the set up parameter values possible for an ODBC Data Source, but we discuss the Server, Translation, and Format tabs parameters in the following sections.

ODBC Data Source Server parameters

The Server set of parameters are important in understanding the impact of some of the SQL functions you perform on the data on the system to which you are connected. Figure 307 shows the Server set of parameters. We also include the Help window for the Commit modes parameter because of the part it plays in transaction integrity.

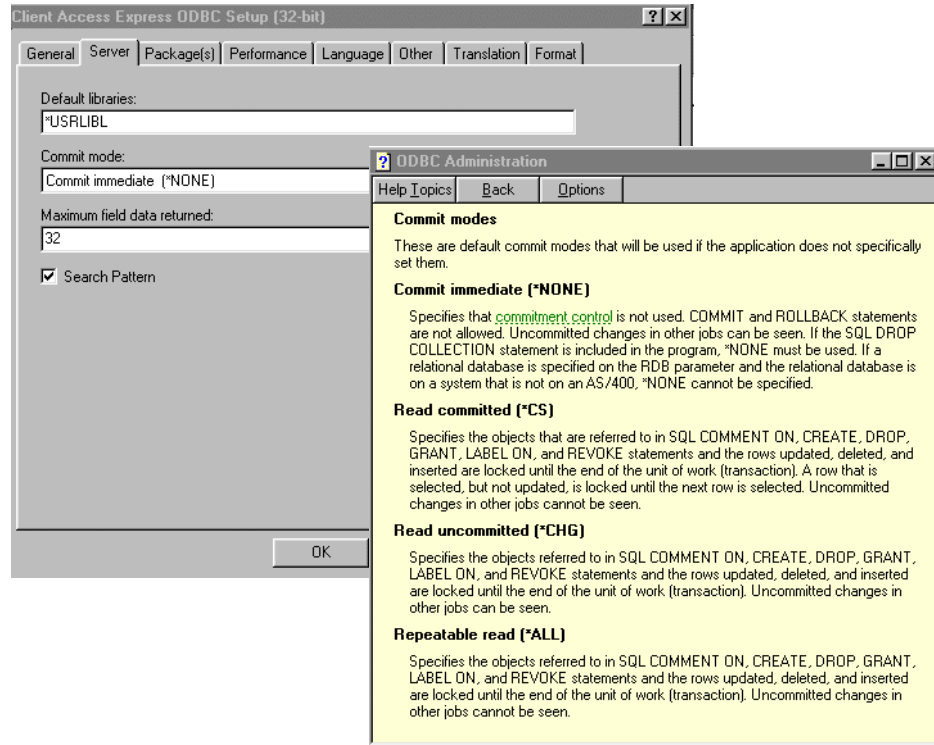


Figure 307. ODBC Data Source: Run SQL Scripts server tab

Default libraries enables you to change the set of libraries available to the user of this ODBC Data Sources. The default (*USRLIBL) means to use the initial library list (INLLIBL) parameter specified on the job description for the OS/400 user profile using this ODBC Data Source.

Commit mode controls the level of DB2 UDB for AS/400 commitment control, including when database changes are considered permanent and whether other users of the same database rows can see column updates that are not yet permanent.

A complete description of commitment control is beyond the scope of this redbook. However, you should understand that in the industry, users of SQL typically expect commitment control to be active. That is, an application design determines what a completed transaction (also called a unit of work) is. Any database row changes (column updates, rows deleted, rows inserted) are not considered permanent until a successful transaction has been completed (transaction boundary). At that time, the application performs a “commit” and all changes are now made permanent. If the application determines that an in-progress transaction should be terminated, it performs a “rollback”. All changes are as if they had never occurred. If the application abnormally terminates before issuing a commit or rollback, the underlying SQL support performs the rollback.

To support commitment control on OS/400, you must also have the tables journaled, and the job using these tables must issue a system operation that starts commitment control for the job. This system operation can be invoked through the OS/400 Start Commitment Control (STRCMTCTL) command or be implicitly invoked by this parameter for values other than *NONE.

A *commit group* is the term used to refer to the rows that are in the process of being updated, deleted, or inserted. As the help text shows objects referred to on COMMENT ON, CREATE, and so forth are also part of this commit group. The commit or rollback applies to all of these rows and objects.

We include the help text here because the OS/400 default is *NONE, which is not generally supported in the industry. This provides a flexible operating environment, such as letting other applications or users access the latest database changes. However, *NONE exposes the table rows, even while being processed by the properly authorized Operations Navigator user, to be modified without a required database Commit/Rollback operation sequence to make any database changes permanent.

For example, using *NONE means any valid SQL statement that changes column data has made a permanent change to the data. If the properly authorized Operations Navigator user mistakenly updates a column using a wrong value for a key, there is no "Rollback" function available to "undo" the change to the wrong row.

You need either a backup copy of the data or an OS/400 journal to recover the original data.

The other commit values specify row locking rules (other applications prevented from updating the same row) and visibility of in-progress changes among applications accessing the same rows.

Maximum field data returned (in KB) specifies the internal buffer size to allocate for a single transfer of data between your client workstation and AS/400 system. If more data than the value specified is to be exchanged, another I/O operation between the AS/400 system and your workstation must be performed to exchange all of the data. The default (32K) is normally the best buffer size for best performance. If you change this size, to a significantly smaller value, you may degrade performance when exchanging large amounts of data, because multiple transmissions will be required.

Search Pattern, when selected, allows the underscore character (_) in AS/400 library and table names to be treated as search patterns (wildcards). When it is not checked, the underscore character is treated as a character. For example, the search pattern is not checked, LIB_A is treated as it is. When checked, LIB_A would include LIB1A or LIB2A and so on. Using an underscore character in an SQL is similar to using a question mark in PC DOS as a single position wildcard, for example: "dir LIB?A /s".

ODBC Data Source Format parameters

Format parameters are important if you have a special operating environment, such as your system requiring country specific or multiple country support. Figure 308 shows you the ODBC Data Source format parameters available to you through Operations Navigator.

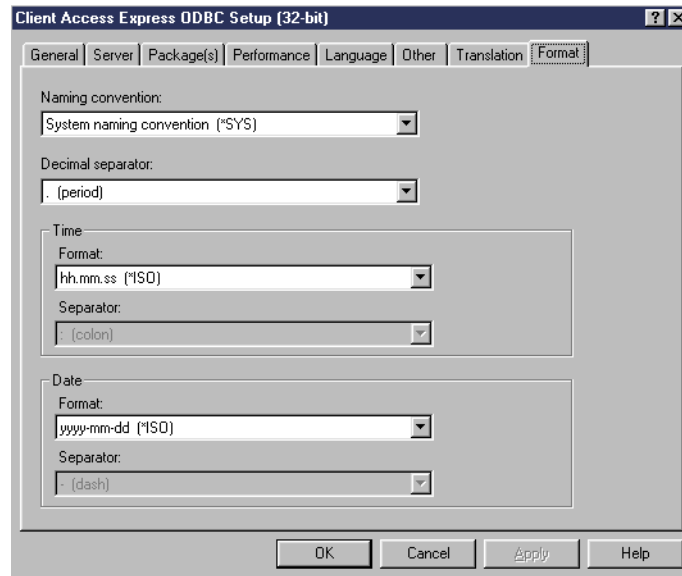


Figure 308. ODBC Data Source: Run SQL Scripts format tab

You must review the online help text to see the details for all of these parameters. These settings shown can be modified. The settings are determined by your requirements.

We include the Format settings here because we do not use the default *SQL naming convention* syntax (includes the period (.) character as a name separator) in some of our examples in this section. We use the (AS/400) System naming convention. This means the forward slash character (/) may be used for this session with As25. This is the normal OS/400 naming convention.

Note: There is an important “operational” difference between using the SQL naming convention and the System naming convention when running SQL statements under Operations Navigator Run SQL Scripts. If you are using the System naming convention and use a non-qualified name, such as a table name with no library qualifier, the system searches for the table within all libraries currently in the session’s (job’s) current library list. If you are using the SQL naming convention, the ANSI standard specification causes the system to look *only* in the *current library* within the session’s current library list.

For example, assume the user portion of the session’s library list is in the order of TEAM02, followed by library TPSTAR02. Also, assume the unqualified table name is CUST_DIM and is stored in library TPSTAR02. Using the SQL naming convention, the system looks for CUST_DIM only in library TEAM02 and does not find it, which results in an error condition. Using the System naming convention, the system first searches the TEAM02 library and then the TPSTAR02 library. Table CUST_DIM will be found, and the SQL statement will run successfully.

ODBC Data Source Translation parameters

In most cases, you never need to view or change the ODBC Data Source translation parameters. This is because your application tables or files are typically stored as using the coded character set identifier (CCSID) numeric value that stores the data according to your national language encoding. In these

cases, any OS/400 data accessed by the client workstation is translated into the appropriate ASCII format as required for viewing or processing on the client.

However, certain OS/400 system files/tables are defined to use the special CCSID 65535. By default, ODBC Data Source processing does not translate data from a file/table with CCSID 65535.

For example, if you want to use Run SQL Scripts against the performance collection files (prefix QAPM...) or a table generated from a Virtual Private Network (VPN) journal (copied to a database file/table), you need to have the character columns translated in most cases. Select the ODBC Data Source **Translate** tab, and select **Translate CCSID 65535**.

For more information on CCSID support, refer to *National Language Support*, SC41-5101.

21.3.6.5 SQL Performance Monitors overview

You can analyze performance of AS/400 SQL statements by putting the appropriate OS/400 job into debug mode and running the SQL statements and viewing the Query Optimizer messages in the job log.

This section describes a more powerful SQL performance analysis tool using V4R4 Operations Navigator. This support provides a graphical interface to IBM-provided SQL queries against data collected by the Memory Resident Database Monitor that was introduced in V4R3. In addition to output equivalent to the debug mode optimizer messages, this monitor can monitor multiple jobs and show the actual SQL statement. This interface is referred to as the *SQL Performance Monitors*.

Before starting an SQL Performance Monitor, you need to determine which job or jobs you want to monitor. There are several techniques you can use to determine the job. We list some of them here:

- If you are using SQL statements running Operations Navigator Database-> Run SQL Scripts, you can click the View option from the menu bar to get a pull-down menu. Clicking Job Log displays your current job's job log. Included in the gray header portion of the job log messages is the name of the job, for example, 139224/QUSER/QZDASOINIT. You can scan down to the earliest job log messages to confirm this job is actually running under the user profile you think it should be.
- If you are not running the job that needs to be monitored, you can get the job name from user of the job, if possible.
- If you know the user profile running the SQL jobs, but do not know which job is the one you want to monitor, you can use the OS/400 Work with Object Locks (`WRKOBJLCK`) command to find the jobs running with that user profile. You may get more jobs than you anticipated. Then, you may need to look in the job logs of each job for SQL-like messages to determine which job or jobs to monitor. For example, you may enter the following command:

```
WRKOBJLCK OBJ(QSYS/TEAM02) OBJTYPE(*USRPRF) MBR(*NONE)
```

We had five jobs: one job name starting with QPADEV000L (5250 emulation), two running Client Access Express database serving with job name starting with QZDASOINIT (not using SSL), and two with the job name starting with QZRCSRVS (central server functions). We looked in the job logs for the two

QZDASOINIT jobs and in one of them found the message 148 rows fetched from cursor CRSR0002.

This QZDASOINIT job was set by Operations Navigator Run SQL Scripts to “include debug messages in job log”.

- You can use the Operations Navigator server jobs interface to find the job by selecting Operations Navigator **Network->Servers->Client Access** to view the Client Access Express servers as shown in Figure 309.

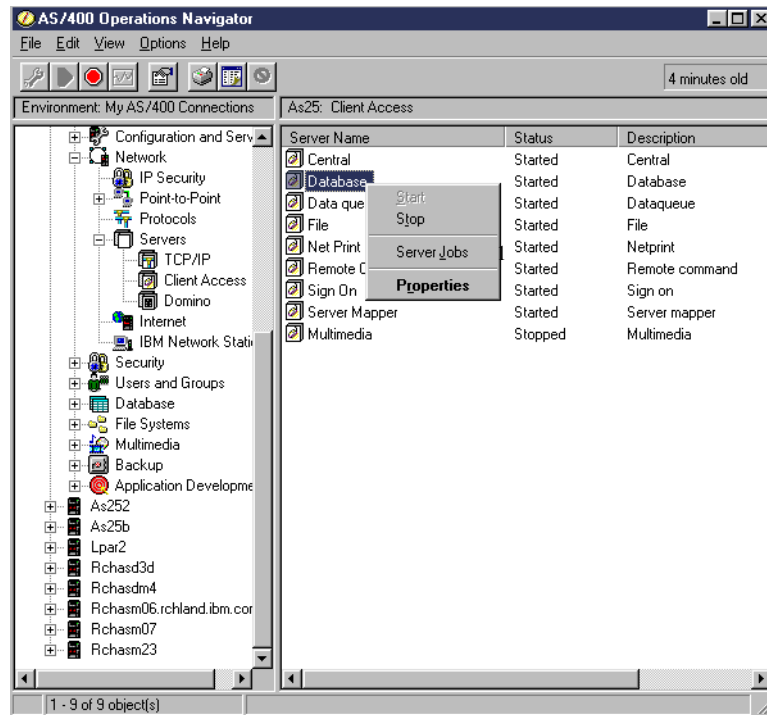


Figure 309. Finding the database server job (Part 1 of 2)

Clicking on Server Jobs brings up a window similar to the one shown in Figure 310 on page 638. This figure shows the database server jobs - QZDASOINIT (not using SSL) that are currently started and shows a current user ID for jobs currently doing active database functions.

Job name	Current user	Server	Job type	Job status	Time entered system	Date entered system	Thread count
Qzdassinit		OS/400 TCP Data	Batch	Printer output	00:42:39	01/25/00	0
Qzdasrvsd		OS/400 TCP Data	Batch	Printer output	00:42:39	01/25/00	0
Qzdassoinit		OS/400 TCP Data	Batch	Printer output	14:45:38	01/28/00	0
Qzdasoinit		OS/400 TCP Data	Batch	Printer output	14:45:38	01/28/00	0
Qzdassoinit		OS/400 TCP Data	Batch	Printer output	14:45:38	01/28/00	0
Qzdasoinit		OS/400 TCP Data	Batch	Printer output	15:13:51	01/27/00	0
Qzdasoinit	QUSER	OS/400 TCP Data	Batch	Active	06:30:38	01/31/00	1
Qzdasoinit	TEAM02	OS/400 TCP Data	Batch	Active	06:30:38	01/31/00	1
Qzdassinit		OS/400 TCP Data	Batch	Printer output	07:12:08	01/03/00	0
Qzdassinit	QUSER	OS/400 TCP Data	Batch	Active	15:05:59	01/28/00	1
Qzdasrvsd	QUSER	OS/400 TCP Data	Batch	Active	15:05:59	01/28/00	1

Figure 310. Finding the database server job (Part 2 of 2)

Figure 310 illustrates an advantage of using the Operations Navigator “servers” support to find a job, compared to using OS/400 5250-display based commands such as the Work with Subsystem Jobs (WRKSBSJOB), Work with Active Jobs (WRKACTJOB), or Work with Object Locks (WRKOBJLCK) commands.

The Operations Navigator interface list the jobs, based on their function. With the OS/400 commands, you need to understand what OS/400 subsystem the server jobs run in and the job name that identifies the server function. In our example, you need to know that the QZDASOINIT jobs do the database serving (in this case ODBC-based) work, and you need to look into the job logs of each active job to find the actual user ID (profile) using the job and lists the user ID on the display.

The OS/400 commands we have discussed show equivalent jobs with the user ID as QUSER. QUSER is the user profile assigned by the system for pre-started Client Access database server jobs. The user profile name actually using the job is indicated in a job log message. The Operations Navigator interface examines the job log messages and shows the active user profile (TEAM02, in our example) if the pre-started job is currently in session with a signed on client.

21.3.6.6 Starting the SQL Performance Monitor example

To run an SQL Performance Monitor, we need to define a new monitor, specify the jobs to be monitored, and specify the data to be collected.

To start the SQL monitoring process, right-click **SQL Performance Monitors**, and select **New...** as shown in Figure 311.

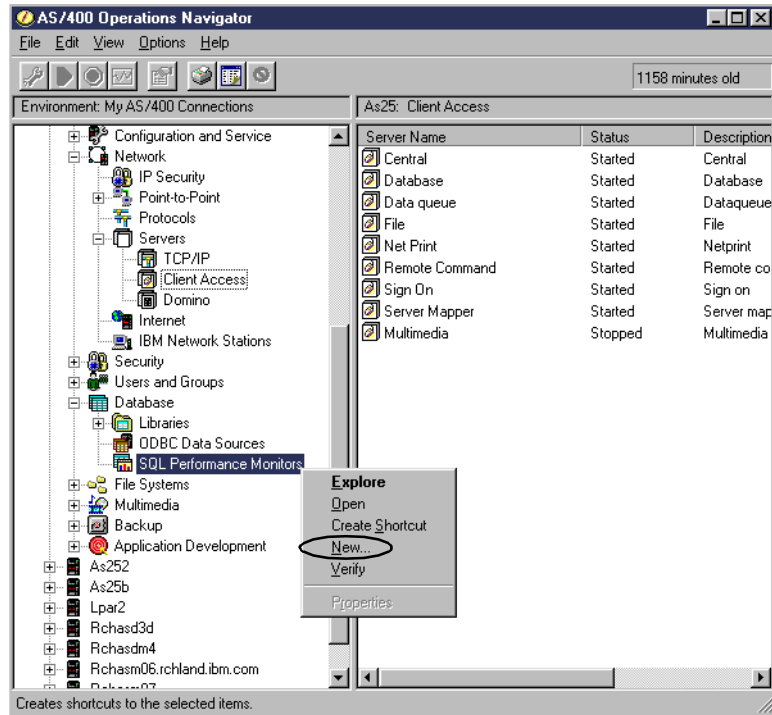


Figure 311. Starting an SQL Performance Monitor (Part 1 of 5)

Select **New...** to bring up the New SQL Performance Monitor dialogue panel with three tabs: General, Monitored Jobs, and Data to Collect.

The General Tab is shown in Figure 312.

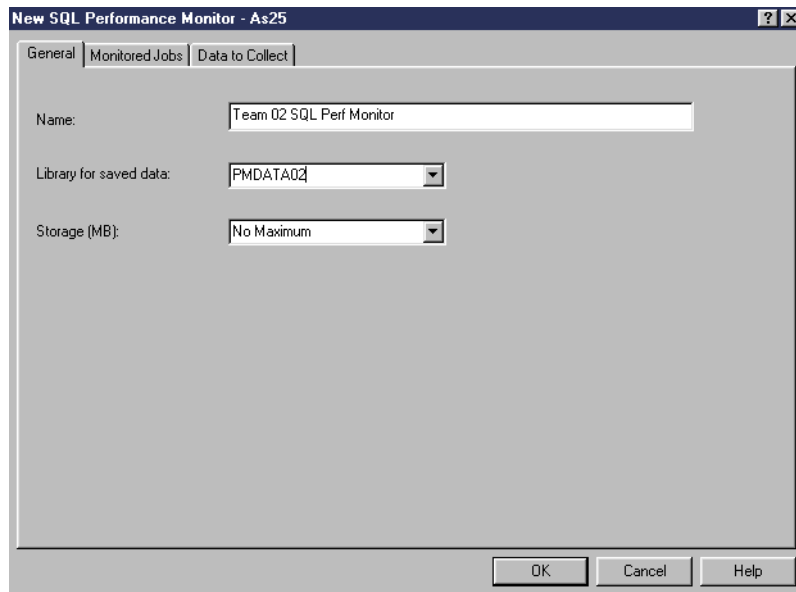


Figure 312. Starting an SQL Performance Monitor (Part 2 of 5)

We already entered the monitor name, the library name that will be used to contain the collected data, and the amount of main storage allocated to the monitoring process.

Do not click the OK button yet, because monitoring all jobs will be started if you have not selected specific jobs under the Monitored Jobs tab. Monitoring all jobs is not recommended on a system with hundreds of active jobs, since the monitoring process can degrade performance.

To specify which OS/400 jobs to manage, click the **Monitor Jobs** tab, which brings up the panel shown in Figure 313.

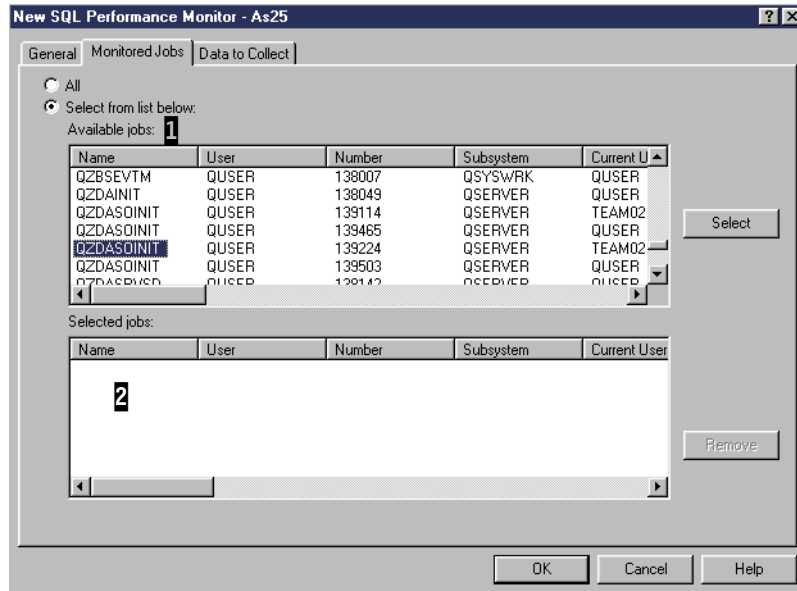


Figure 313. Starting an SQL Performance Monitor (Part 3 of 5)

You can select to monitor all jobs or to select jobs from the Available jobs list pane shown in 1. As you select a job and click the **Select** button, the job information is entered into the Selected jobs list pane 2. You remove the selected jobs by selecting a job in the Selected jobs pane and clicking the **Remove** button.

In this example, we already scrolled down the active job names to display the ones shown in 1. We select to monitor only job QZDASOINIT/QUSER/139224 - with TEAM02 as the current user. We recommend that you monitor as few jobs as possible, because monitoring a large number of active jobs can impact normal productivity.

When you finish selecting jobs, click the **Data to Collect** tab. This brings up the panel shown in Figure 314.

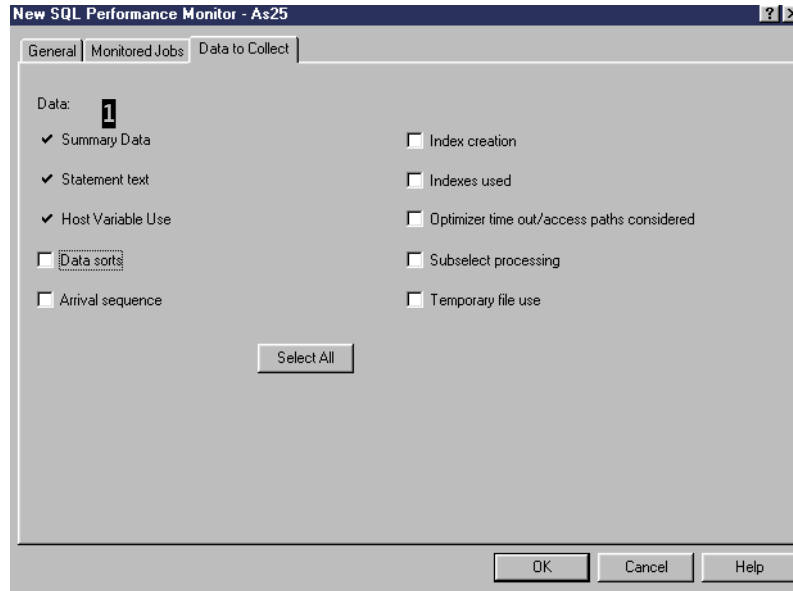


Figure 314. Starting an SQL Performance Monitor (Part 4 of 5)

This panel shows three sets of SQL monitor data collected during every monitor collection period at **1**. You can specifically include other sets of data or simply click the **Select All** button. You should “select all”, unless you understand the application implementation in detail so that you need to collect only specific information.

When you are satisfied with your monitor collection specification, click the **OK** button to return to the original SQL Performance Monitor window that shows the monitor status on the right pane in Figure 315.

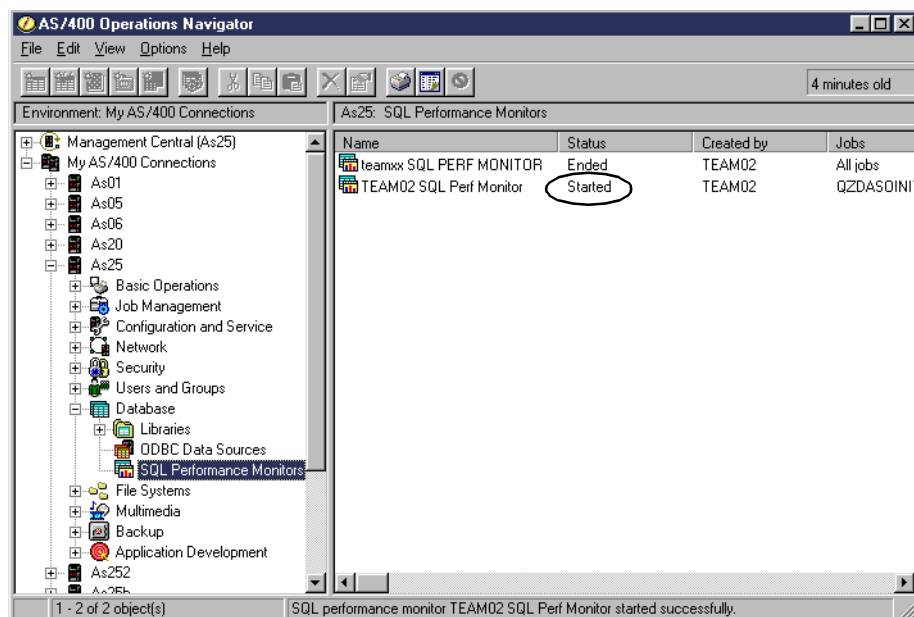


Figure 315. Starting an SQL Performance Monitor (Part 5 of 5)

In our example, we use Run SQL Scripts to run the SQL statement. This statement has a relatively complex WHERE clause as shown in Figure 316.

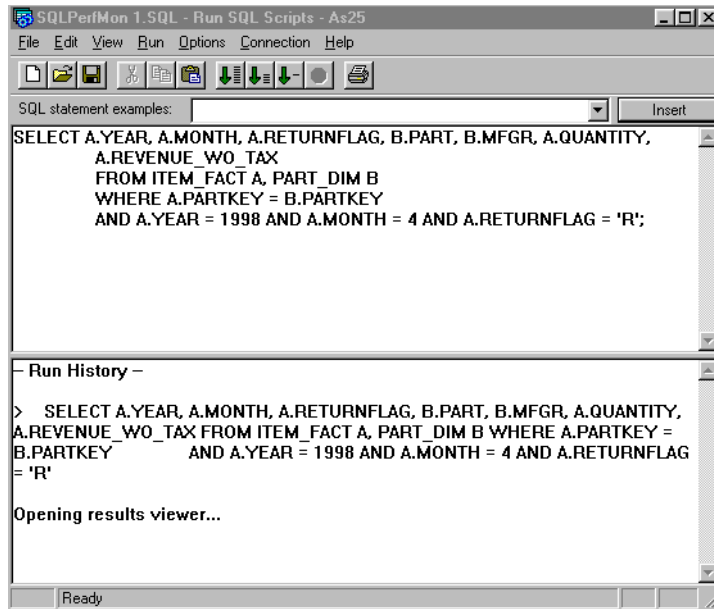


Figure 316. SQL Performance Monitor: SQL statement monitored

Operations Navigator Run SQL Scripts support uses ODBC support. In our example (Figure 316), the SQL statement has already been run as evidenced by its appearance within the “Run History” pane. The message *Opening results viewer...* indicates that the results of the SQL select statement have already been displayed to the Operations Navigator user.

The SQL Performance Monitor can monitor all SQL work performed on OS/400. In addition to Operations Navigator Run SQL Scripts jobs, other users of OS/400 SQL support would include a client workstation Visual Basic program accessing the OS/400 via ODBC, a client workstation Java applet accessing the OS/400 via JDBC, a local AS/400 program using embedded SQL in the RPG, COBOL, or C program, a local AS/400 program using the SQL CLI (Call Level Interface) in RPG, COBOL, C, or Java.

OS/400 also has a 5250 workstation-based SQL interface running under the Start SQL (STRSQL) command.

21.3.6.7 Reviewing the SQL Performance Monitor results

The SQL Performance Monitor statistics are kept in main storage for fast recording, but need to be written to database files to use the Operations Navigator interface to review the results. You can get the statistics written to database files by either “pausing” or “ending” the monitor.

Right-click on the active SQL Performance Monitor, and a pop-up window appears that lists Pause, End, and other monitor actions as shown in Figure 317.

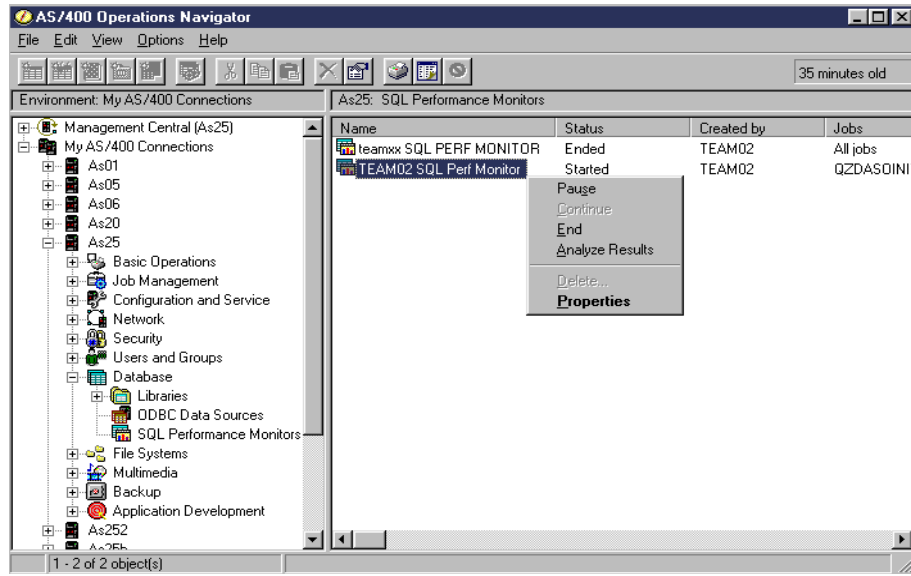


Figure 317. Managing the SQL Performance Monitor

The possible managing functions are:

- **Pause:** This function stops the current collection of statistics and writes the current statistics into several database files/tables that can be queried by selecting the Analyze Results action. The monitor remains ready to collect more statistics, but requires the Continue action to restart collection.
- **Continue:** This function restarts the collection of statistics for a monitor that is currently “paused”.
- **End:** This function stops and ends the monitor and writes the current collection of statistics to the database files or tables.
- **Analyze Results:** This brings up a window with three tabs for selecting ways to look at (query) the collected statistics in the database files or tables:
 - Summary Results
 - Detailed Results
 - Composite View

We show an example of a Detailed Result query report in Figure 321 on page 646.

- **Properties:** This function brings up a window with three tabs representing the original monitor definition:
 - General
 - Monitored Jobs
 - Saved Data

We show an example of Saved Data tab files in Figure 318 on page 644.

Figure 318 on page 644 shows an example of the monitor **Properties->Saved Data** tab panel for our monitor.

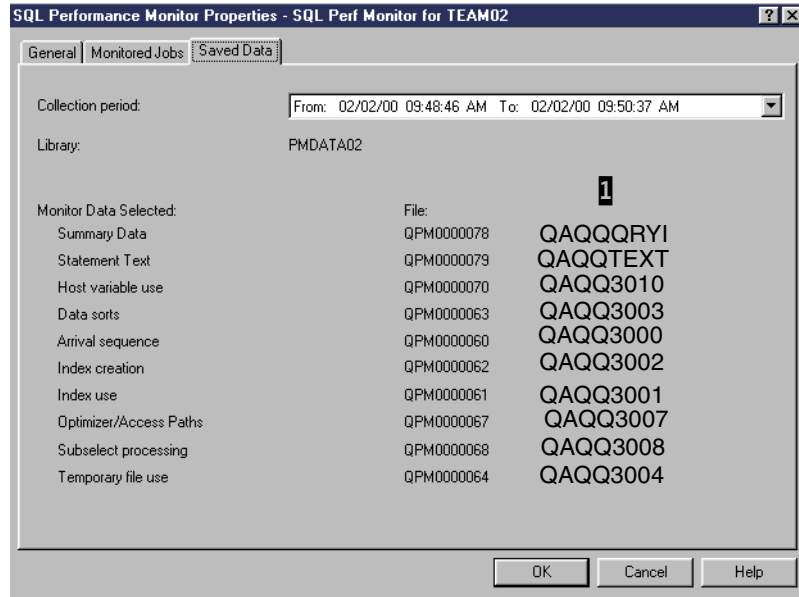


Figure 318. SQL Performance Monitor: Properties

Figure 318 shows the files or tables that correspond to the Data to Collect, specified in Figure 314 on page 641. The monitor files have generic names with sequence numbers as suffixes. We show the corresponding file or table name (under **1**) described in the memory resident Database Monitor documentation in *DB2 UDB for AS/400 SQL Programming*, SC41-5611.

The SQL Performance Monitor file name numeric suffix is updated when each monitor is started.

Analyzing SQL Performance Monitor results

OS/400 provides many pre-defined queries to view the recorded statistics. You can select these queries by checking the various query types on the Analyze Results panels. To begin viewing the results, right-click the “paused” or “ended” monitor. Select **Analyze Results** from the pop-up window.

Figure 319 shows the first results panel that groups queries according to three tabs:

- Summary Results
- Detailed Results
- Composite View

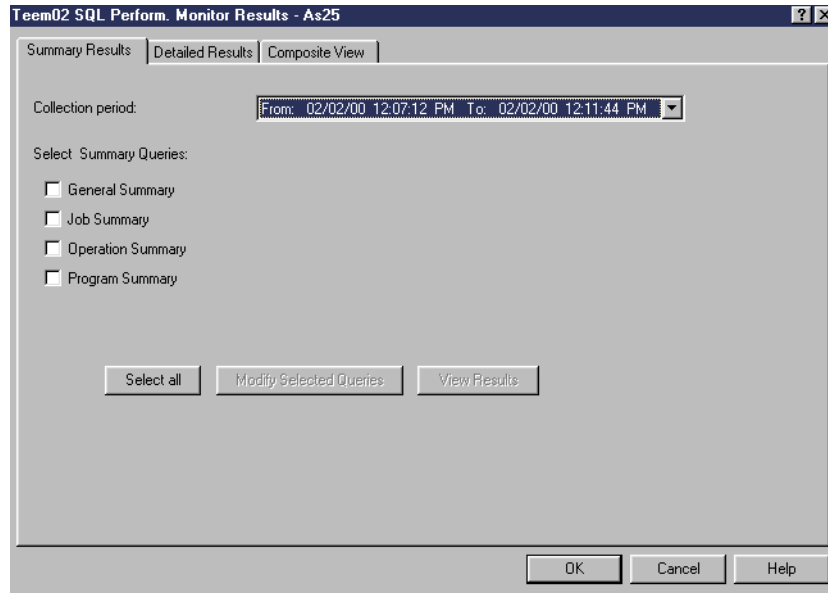


Figure 319. SQL Performance Monitor - Analyze Results: Summary Results queries

Table 50 on page 647 summarizes the various IBM-provided queries under each tab. You can select individual queries or use the Select All button. After selecting the queries you want to run, select the **View Results** button, which will be activated.

You can even choose to modify the pre-defined queries and run the “new” queries, by selecting the Modify Selected Queries button.

An in-depth discussion of using the SQL Performance Monitor results to improve performance is beyond the scope of this redbook. However, we show a sample of the query results output for our SQL statement used in Figure 316 on page 642.

To obtain the query results shown in Figure 321 on page 646, you must first select the **Detailed Results** tab on the Performance Monitor Results window shown in Figure 319. This brings up the Detail Results panel shown in Figure 320 on page 646.

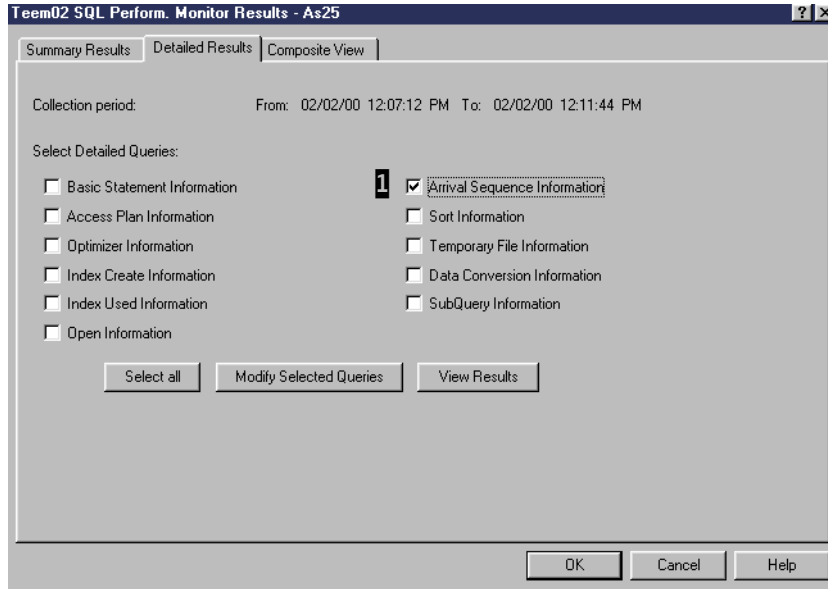


Figure 320. SQL Performance Monitor: Detail results

You can select individual detail query reports, “select all” queries, and even modify the provided queries. When finished selecting the queries you want, click the **View Results** button.

The OS/400 query optimizer support includes an Index Advisor function. This support includes, when appropriate, a recommendation that a new index should yield improved performance. Columns that should be used in the index are listed. To view this detailed information, you must first select to view **Arrival Sequence Information** as shown at **1** in Figure 320. Click the **View Results** button to access a panel similar to the one shown in Figure 321.

Figure 321. SQL Performance Monitor: Arrival Sequence Information (Index Advisor function)

To show the information in Figure 321, we had to scroll to the right to find the Advised Index and Advised Index Keys columns shown at **1**. You can see at **2**, we compressed several columns in the results to make the index path information fit within the window.

The lab exercise that can be downloaded to your AS/400 system on the PC workstation listed in the beginning of this section. Use the “Self study lab” to familiarize yourself with the power of the SQL Performance Monitor, as well as most of the Operations Navigator - Database support. The lab also includes tips on tuning SQL performance.

Table 50 summarizes the results queries grouped under the Summary Results, Detailed Results, and Composite View tab results categories.

Table 50. SQL Performance Monitor: Analyze results queries

Query group or name	Description
Summary Results Group	Group contains several views of summary information
General Summary	Contains a row of information that summarizes all SQL activity
Job Summary	Contains a row of information for each job. Each row summarizes all SQL activity for that job
Operation Summary	Contains a row of summary information for each type of SQL operation
Program Summary	Contains a row of information for each program that performed an SQL operation. Each row summarizes all SQL activity for that program
Detailed Results Group	Group contains several views of detail level information
Basic Statement Information	Summarizes the activity for each unique SQL statement
Access Plan Information	Contains a row of information for each SQL statement that required the access plan (algorithm to find rows) to be rebuilt
Optimizer Information	Contains a row of optimizer information for each subselect in an SQL statement
Index Create Information	Contains a row of information for each SQL statement that required an index to be built
Index Used Information	Contains a row of information for each SQL statement that needed to use the index
Open Information	Contains a row of information for each table open activity for each SQL statement
Arrival Sequence Information	Contains a row of information for each select that required rows to be processed in arrival sequence order (if appropriate, includes column names for a recommended new index)
Sort Information	Contains a row of information for each sort that an SQL statement performed
Temporary File Information	Contains a row of information for each SQL statement that required a temporary result
Data Conversion Information	Contains a row of information for each SQL statement that required data conversion
SubQuery Information	Contains a row of subquery information
Composite View	These reports join data from selected detail reports and includes summary data and SQL statement text
Arrival Sequence	Contains the table scan data for monitored jobs
Data Sorts	Contains details of data sorts that monitored jobs performed
Host Variable Usage	Contains the host variable values used by monitored jobs

Query group or name	Description
Optimizer time outs and access paths	Contains details of any query time outs in monitored jobs
Indexes Used	Contains details of how indexes are used by monitored jobs
Index Creation	Contains details of index creation by monitored jobs
Subselect Processing	Contains information about each subselect in an SQL statement
Temporary File Use	Contains details of temporary files that monitored jobs created

21.3.7 Database performance analysis: Centerfield Technologies Tools

Centerfield Technology's Advanced Performance Server (APS) provides the infrastructure to analyze and tune the AS/400 system for high performance access. The APS is designed for today's leading SQL-based enterprise resource planning, customer relationship management, business intelligence, and Web-based applications.

The Advanced Performance Server's architecture meets the demands of enterprise-wide SQL-based applications. Unlike other AS/400 performance management products, it optimizes the database and the AS/400 query engine. The server facilitates index management, database activity tracking, and analysis, and gives you insight into overall system performance from a database perspective.

The Centerfield Technology's Advanced Performance Server allows Customers to:

- Visually develop, understand, and tune SQL statements
- Intelligently manage and create database indexes for optimum performance
- Track users, database activity, and queries
- Analyze performance data to pinpoint opportunities

It features the following functions:

- Data Base Profiler
- Index Advisor
- Performance Tuning & Management
- Historical Tracking & Analysis

Some sample windows of the Advanced Performance Server are shown and explained in the following figures and text.

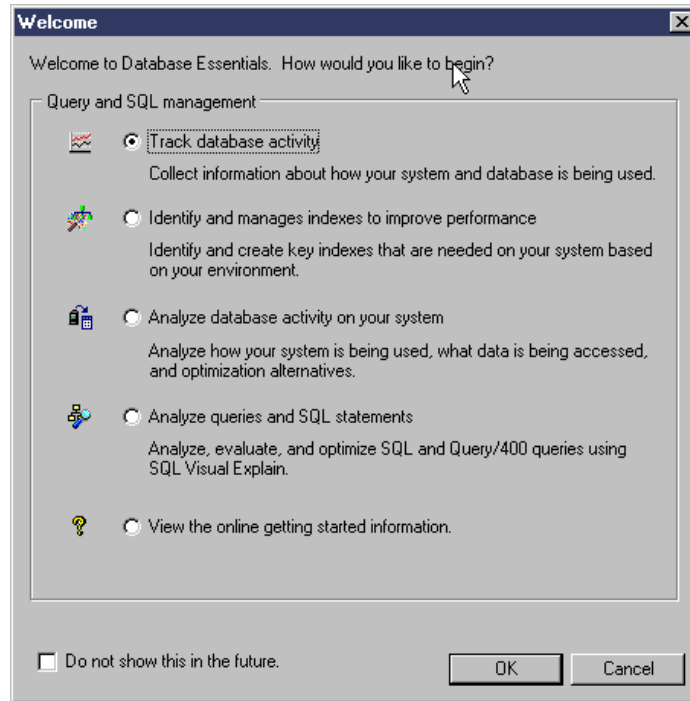


Figure 322. Query and SQL management

The Welcome display (Figure 322) allows you to select the various tasks to invoke:

- Collect database information
- Identify and create indexes
- Analyze database activity
- Analyze queries and SQL statements

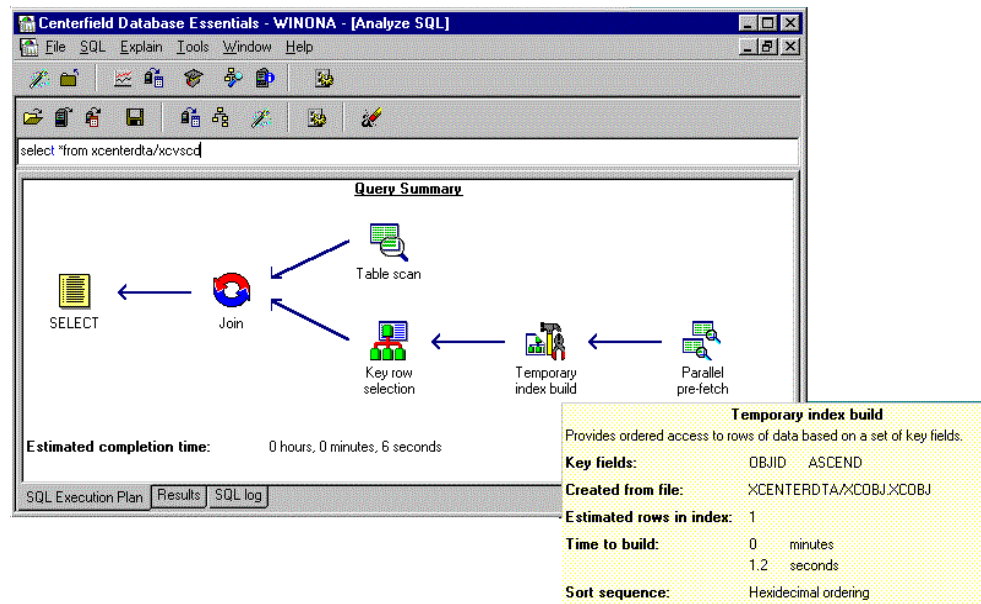


Figure 323. Analyze SQL requests

The SQL analysis (Figure 323 on page 649) gives a graphical overview of all components being used of a particular SQL request before it is actually executed. Each step is represented by a unique icon and contains context-sensitive help so statements can be easily learned and understood. It also surfaces the numerous tuning options available on the AS/400 system so you can experiment to see what effect those options can have on performance.

In the previous example, the SQL statement uses three indexes. Joining them together, the results are then grouped by and sorted before they are returned to the application.

As you can see from the statement above, SQL allows you to specify the data you want, how you want it formatted, and so on. How the data is actually retrieved is hidden and taken care of by the query optimizer.

To diagnose performance problems and fine tune your environment, it sometimes becomes necessary to look under the covers to see what is happening. This requires:

- The ability to look inside the query optimizer decisions
- The ability to understand what you see
- The ability to take corrective action

The Advanced Performance Server helps you with all three of these requirements. SQL Visual Explain and its associated documentation are primarily available for this purpose.

Recommended index type	File	Library	Recommended key fields	Number of creates
Binary radix	F0311	FNPLIBF	RPAN8 ASC, RPDOC ASC, RPDCO ASC, ...	178
Binary radix	F43121	FNPLIBF	PRLOTN ASC	14
Binary radix	F4101	FNPLIBF	IMSRP2 ASC	34
Binary radix	F41021	FNPLIBF	LILOCN ASC, LIPQOH ASC	32
Binary radix	F43092	FNPLIBF	PXDCCO ASC, PXDCTO ASC, PXMCO ASC, ...	16
Binary radix	F0311	FNPLIBF	RPAN8 ASC, RPDIVJ ASC	2
Binary radix	F4108	FNPLIBF	IORLOT ASC, IOLOT1 ASC, IOLOTN ASC	13
Binary radix	DMAACPP	DMLIBF	AAACST ASC, AAOST ASC	2
Binary radix	F43092	FNPLIBF	PXDCCO ASC, PXMCO ASC, PXDCCO ASC, ...	12
Binary radix	F43092	FNPLIBF	PXMCO ASC	4
Binary radix	F00950	JDFSECTW	FSSEY ASC, FSUSER ASC, FSOBNM ASC, ...	45
Binary radix	F43121	FNPLIBF	PRDCCO ASC, PRMCO ASC	7
Binary radix	IBMSN00001	DPRDLY	TRY_C00001 ASC	108
Binary radix	F4101	FNPLIBF	IMITM ASC	2
Binary radix	F4101	FNPLIBF	IMSRP2 ASC, IMITM ASC	1
Binary radix	F43121	FNPLIBF	PRAN8 ASC, PRMCO ASC, PRDCTO ASC, ...	5
Binary radix	F43092	FNPLIBF	PXDCCO ASC, PXDCTO ASC, PXMCO ASC, ...	4
Binary radix	F43092	FNPLIBF	PXDCCO ASC, PXMCO ASC	6
Binary radix	F43121	FNPLIBF	PRMCO ASC, PRDCCO ASC	6
Binary radix	F43121	FNPLIBF	PRDCCO ASC, PRDCTO ASC, PRKCOO AS, ...	2
Binary radix	IBMSN00003	DPRDLY	PRIORITY ASC, COPY_OWNER ASC, COPY_TAB, ...	54
Binary radix	F4101	FNPLIBF	IMPRP1 ASC	6
Binary radix	F4111	FNPLIBF	ILITM ASC, ILMCO ASC, ILTRDJ DESC	6
Binary radix	F4102	FNPLIBF	IBSRP2 ASC	2
Binary radix	F0411	FNPLIBF	RPAN8 ASC, RPDIVJ ASC	14
Binary radix	F4108	FNPLIBF	IORLOT ASC, IOLOT1 ASC	3
Binary radix	F43121	FNPLIBF	PRDCCO ASC, PRMCO ASC, PRLOTN ASC	3
Binary radix	F98DRSUB	JDFSECTW	DRPUBDS ASC, DRPUBHST ASC, DROBNM, ...	14
Binary radix	F43099	FNPLIBF	POKCOO ASC, PODCCO ASC, PODCTO AS, ...	30

Figure 324. Index advisor

The index advisor (Figure 324) automatically analyzes collected activity looking for index changes that are needed on your system. Based on that activity, it uses a process known as *index melding* to determine the optimal set of indices needed for best overall performance.

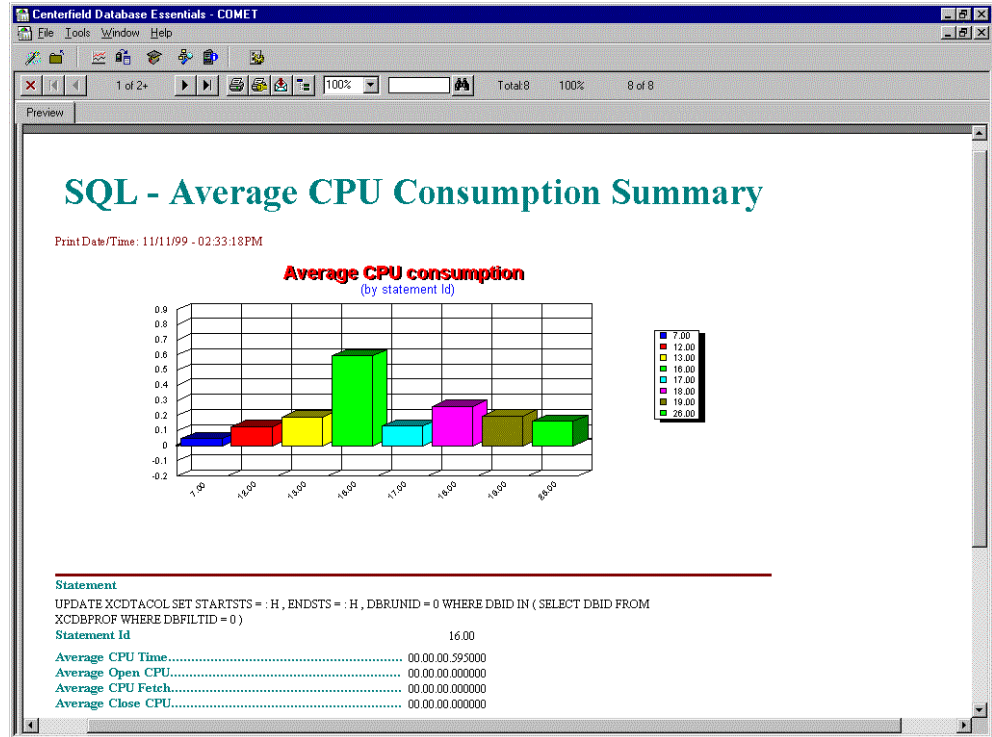


Figure 325. Visual explanation

Database Analysis Manager allows you to pinpoint opportunities for improvement. You are given the ability to analyze performance data so you can focus on the 20% of the queries that can give you the 80% of the benefit.

Note

More information regarding the database analysis tools from Centerfield Technologies can be found at: <http://www.centerfieldtechnology.com>

You can also request more information by sending e-mail to:
info@centerfieldtechnology.com

21.3.8 Other AS/400 performance analysis tools

The following sections describe some of the other performance analysis tools that can be used depending on the situation and requirements.

21.3.8.1 Job trace

The job trace provides information on the program modules called when a job is running. It collects performance statistics for the specified job. A trace record is generated for every external program call and return, exception, message, and workstation wait in the job. At least two, and usually more, trace records are generated for every I/O statement (open, close, read, and write) in a high-level language program.

Tracing has a significant affect on the performance of the current job. Time stamps shown may indicate longer processing within a program than actually

occurred. However, you can use the time stamp (system clock) for the relative execution time compared to other programs or modules listed in the trace.

The job details of the specific AS/400 job (QZDASOINIT) serving the particular run of the client/server application must be determined before attempting to start a job trace. You can determine the correct fully qualified QZDASOINIT job name by using the WRKCFGSTS command or the WRKDEVSTS command that names the client workstation. Or, you can run the WRKOBJLCK command and specify the client workstation's signon user profile and object type *USRPRF.

The Job Trace collection includes the invocation of Debug. It collects the SQL optimizer messages as described in the previous section. The steps used to collect job trace are:

1. Start Service Job (STRSRVJOB) by using the following command:

```
STRSRVJOB JOB (number/user/name)
```

2. Start Debug:

```
STRDBG UPDPDPROD(*YES)
```

3. Start Trace:

```
TRCJOB MAXSTG(16000) TRCFULL(*STOPTRC)
```

4. Change Job to produce job log:

```
CHGJOB JOB (number/user/name) LOG(4 00 *SECLVL)
```

5. To end the job trace, debug, and service job, use the following commands:

- TRCJOB SET(*OFF) to end the trace and print a report
- ENDDBG to exit debug mode
- ENDSRVJOB to end the service job

21.3.8.2 Performance Explorer (PEX) STATS

PEX STATS can give you an extremely accurate account of which programs, processes, and tasks are using AS/400 server CPU and disk I/O resources. PEX STATS can provide extremely useful information on the high-level language programs (such as for WorldSoftware applications) and SQL queries for interactive and batch jobs. You may collect this data for a set of jobs or for the entire system.

The activity sequence for collecting PEX STATS data is shown here:

1. Define a PEX statistics definition by using the ADDPEXDFN command. Specify the following statements:

```
TYPE (*STATS), JOB(job_number/job_user/job_name)  
DTAORG(*HIER)
```

Note

You may select all the AS/400 system jobs by specifying JOB(*ALL). When collecting PEX STATS data for all the jobs, specify *FLAT (instead of *HIER) for the data organization, DTAORG parameter.

2. Start a PEX collection session by entering the STRPEX command. Specify your STATS PEX definition.

3. Let the PEX collection run for one half to one hour. Then, end PEX collection by using the `ENDPEX` command. You can replace or add to data already collected under the same `DTAMBR` name with the `RPLDTA` parameter. You can also supply a meaningful description of the collected data, for example:

```
ENDPEX SSNID(active session name) DTALIB(your library name) DTAMBR(*SSNID)
RPLDTA(*YES) TEXT('JDE WORD Batch Processing')
```

PEX STATS is primarily a program-oriented performance tool, not a job-oriented tool. It captures precise data on each program's resource usage by interrogating resource counters at each program-to-program transition. In the OneWorld environment, use PEX STATS to determine which programs are called the most, use high CPU time, or perform large disk I/Os.

If the data is collected for a single job and `DTAORG(*HIER)` is used, you can trace the job in terms of Caller-Called programs. For details on how to print and interpret PEX STATS reports, refer to Chapters 3 and 4 of *AS/400 Performance Explorer Tips and Techniques*, SG24-4781.

21.3.8.3 Performance Explorer Trace

PEX Trace can provide useful information in the areas such as identification of SETOBJACC candidates, objects involving large number of I/Os, objects/I/Os causing unbalanced disk arm utilization, and much more. For a complete list, refer to Section 7.1 of *AS/400 Performance Explorer Tips and Techniques*, SG24-4781. Collecting and analyzing the PEX Trace data using the Enhanced PEX Trace function is described in Chapters 7 and 9 of *AS/400 Performance Explorer Tips and Techniques*, SG24-4781. You can download the V4R3 and V4R4 versions of the SMTRACE and QYPINT libraries from the Web site at:

http://www.as400service.ibm.com/p_dir/pexgui.nsf

PEX Trace collection and analysis can also be done by using a new graphic interface tool, PEXGUI. This tool can also be downloaded from the same Web site.

21.3.8.4 OneWorld journal file analysis

To identify all OneWorld files currently being journaled, print the journal attributes for the journal by using the following command:

```
WRKJRNA JRN(B732xCTL/OW_JRNL) OUTPUT(*PRINT)
```

To identify all the journal receivers to evaluate their size and ASP, run the following command:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*JRNRCV)
```

Enter option 5 to display description of each journal receiver on the system.

21.3.8.5 OneWorld index analysis

The availability of table indexes has an affect on the behavior of OneWorld applications in general. It is valuable to record all OneWorld table indexes for the environment being evaluated. Use the Display Database Relationship (`DSPDBR`) command for all the files in the Object Librarian library, Control Tables library, and the Business Data library. If you are a coexistence customer, use this command for the Common Library Name library as well, for example:

```
DSPDBR FILE(Object Librarian Library/*ALL)
```

21.3.9 Client Access Express and client application PC tools

This section describes performance analysis and reporting for a PC client.

21.3.9.1 ODBC trace

An ODBC trace captures ODBC connection information, ODBC API calls, SQL statements, and ODBC error messages. An ODBC trace is useful in both error diagnostics and performance tuning. The trace occurs at a level between the application and the ODBC driver. Therefore, it is helpful in showing what is being passed to the ODBC driver from the application. ODBC 3.0 includes its own trace utility.

To perform an ODBC trace, complete these steps:

1. Click **Start->Programs->IBM AS/400 Client Access Express->ODBC Administrator**.
2. Select the **Tracing** tab, and click **Start Tracing Now** (Figure 326).
3. Recreate the error.
4. Select the **Tracing** tab again. Click **Stop Tracing Now** to end the trace. Do not leave the trace running since it will have a severe performance impact on ODBC jobs.

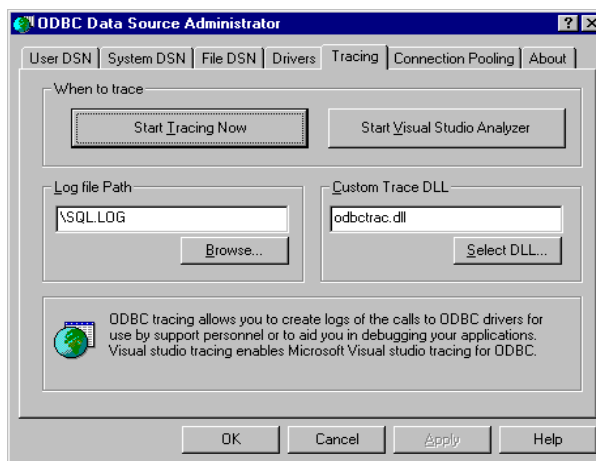


Figure 326. ODBC Data Source Administrator Tracing tab dialog

21.3.9.2 PC error messages

Record the PC application error message, the Client Access ODBC error message, and the SQL State.

The PC application provides an error message. Include the exact text and error numbers in the error message. When a printer is unavailable, the image of the error message can be captured by pressing the Print Screen key. Print Screen copies a bitmap of the current display to the clipboard. Use your preferred application to access and store the error bitmap in a folder for future reference.

21.3.9.3 Client Access ODBC error messages

The Microsoft ODBC specification defines a standard, layered error handling protocol. Error messages include the identity of the error source, the actual error text, the ODBC SQLState, and the return code. ODBC error messages have the following format:

[vendor] [ODBC-component] [data-source] error-message text

The prefixes in brackets identify the vendor issuing the error, the ODBC component involved, and the data source (DBMS). Client Access ODBC uses the following format to identify errors whose source is the Client Access ODBC driver or DB2 UDB for AS/400:

[IBM] [Client Access ODBC Driver (32-bit)] [DB2/400 SQL]

The prefixes that are used in Client Access ODBC driver error messages are listed and explained in Table 51.

Table 51. Prefixes used in Client Access ODBC driver error messages

Prefix	Explanation
[IBM]	Identifies IBM as the vendor issuing the error. If this value is something other than IBM, the failure did not occur in IBM code. An error starting with [Microsoft] [ODBCxxx DLL] implies that the failure was at the MS ODBC driver manager manager level.
[Client Access ODBC Driver (32-bit)]	Identifies the Client Access ODBC driver as the component reporting the error. When Client Access ODBC reports the error but the [DB2/400] prefix does not follow, the error occurred on the PC. The request was never sent to the AS/400 system.
[DB2/400]	Identifies the AS/400 system as the data source. This field, when included, indicates that the failure originated on the AS/400. The AS/400 job log may contain additional information for these types of errors.

21.3.9.4 Client Access service packs

Updates to the Client Access ODBC driver are included in the Client Access service packs. Verify that the latest Client Access service pack is applied on the PC. Service packs can be ordered through ECS or downloaded from the Client Access home page on the Web: <http://www.as400.ibm.com/clientaccess>

21.3.9.5 JDEDEBUG.LOG

JDEDEBUG.LOG is not only a comprehensive log of what is happening. Since everything is time stamped, it is also a useful performance analysis tool. Simply looking for the function being performed that caused xx seconds to pass until the next function is quite valuable.

In addition to looking at the database from the AS/400 side, it can be useful to look at the SQL statements from the PC side. The Advanced Technologies World Server has two tools designed to analyze the number and type of SQL statements at the table level. One tool looks at the JDEDEBUG.LOG file. The other tool looks at the SQL.LOG file produced by an ODBC SQL trace.

21.3.9.6 Server Administration Workbench (SAW)

You can use the OneWorld Server Administration Workbench (SAW) monitor programs to observe the number of OneWorld network connections and other load-based factors that may be used to evaluate the performance of server processes. This information is useful to make load balancing decisions, such as increasing the number of processes or network connections to increase

performance. You can also use the monitoring programs to enable various logs and traces useful for troubleshooting purposes.

This SAW application provides the server administrator with vital statistics concerning the internal functions of OneWorld. The functionality of SAW includes embedded notification capabilities. Using configurable e-mail or Pager push delivery mechanisms, specified server administrators are notified:

- Every time a server is started or stopped
- Of initial and changed server status, such as up or down
- Of overloaded conditions when a process' configurable outstanding request threshold is reached

The SAW program also allows you to do the following tasks:

- View the number of active JDENET and OneWorld processes running
- View incoming processing requests that are issued by a OneWorld process
- Shut down OneWorld running on the server
- View workstation and server connection information
- View the OneWorld workstation and server configuration settings
- Notify workstations if there is a server shut down
- Allow the collection and display of OneWorld message-flow statistics

SAW can be accessed on the AS/400 system by signing on as ONEWORLD and typing *SAW* on the command line (Figure 327).

```
SAW                      OneWorld Server Administration Workbench          J.D. Edwards
JDECS4:  ONEWORLD                Port 6008                      03-21-00 19:03:34

Select option:

    1. Work with Servers
    2. Work with Server Processes
    3. Work with Server Resources
    4. Work with Server Log files
    5. Work with Server INI file

Selection
====>

F3=Exit
```

Figure 327. Server Administration Workbench

From here, on the command line, simply type the number associated with the function you want to perform.

There is also a Server Administration Workbench application that can be accessed from a workstation that has OneWorld installed on it. To access the SAW application in this case, follow these steps:

1. From the OneWorld Explorer, access Fast Path, and type GH9011 to reach the System Administration Tools menu (Figure 328).

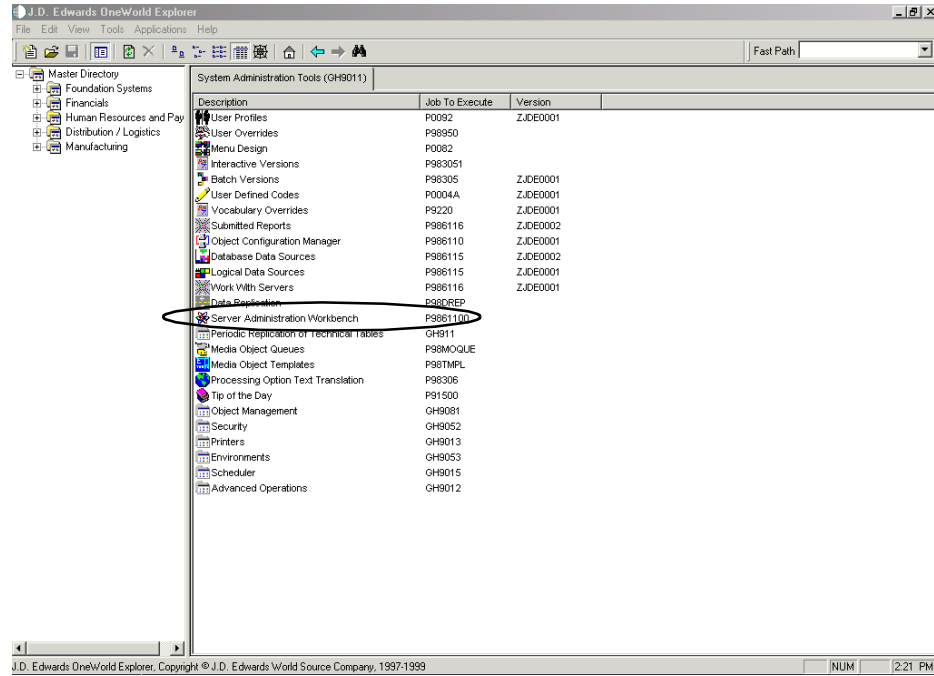


Figure 328. System Administration Tools

2. Double-click the **Server Administration Workbench** application (P9861100).
3. Enter your AS/400 name in the Server Name field, and then click **OK** to access the display shown in Figure 329 on page 658.

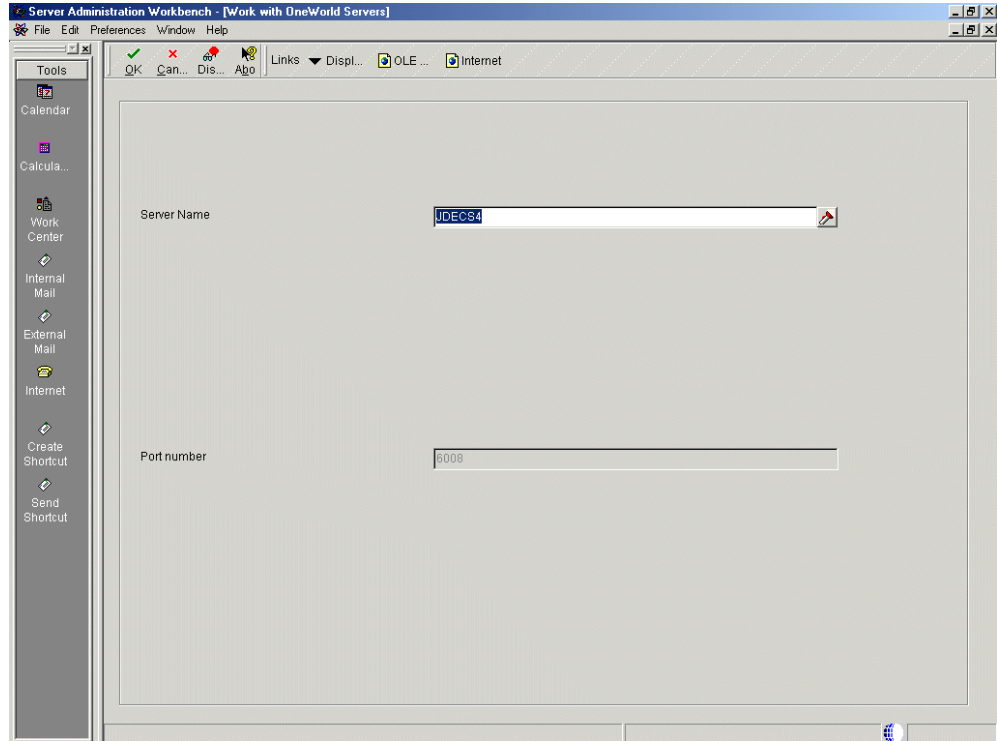


Figure 329. Work with OneWorld servers

4. To look at logs, in the Work with Server Log files folder, double-click the **Work with Server Log files** item to access the display shown in Figure 330.

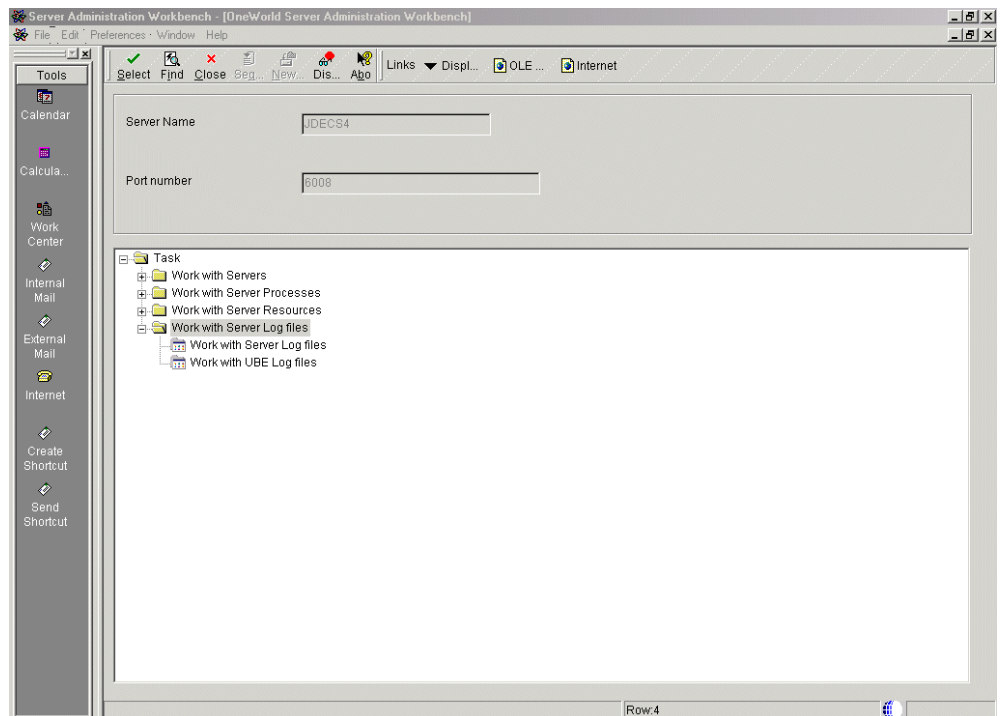


Figure 330. Work with Server log files

- Highlight the appropriate log file, and click **View Log File** on the Row menu (Figure 331).

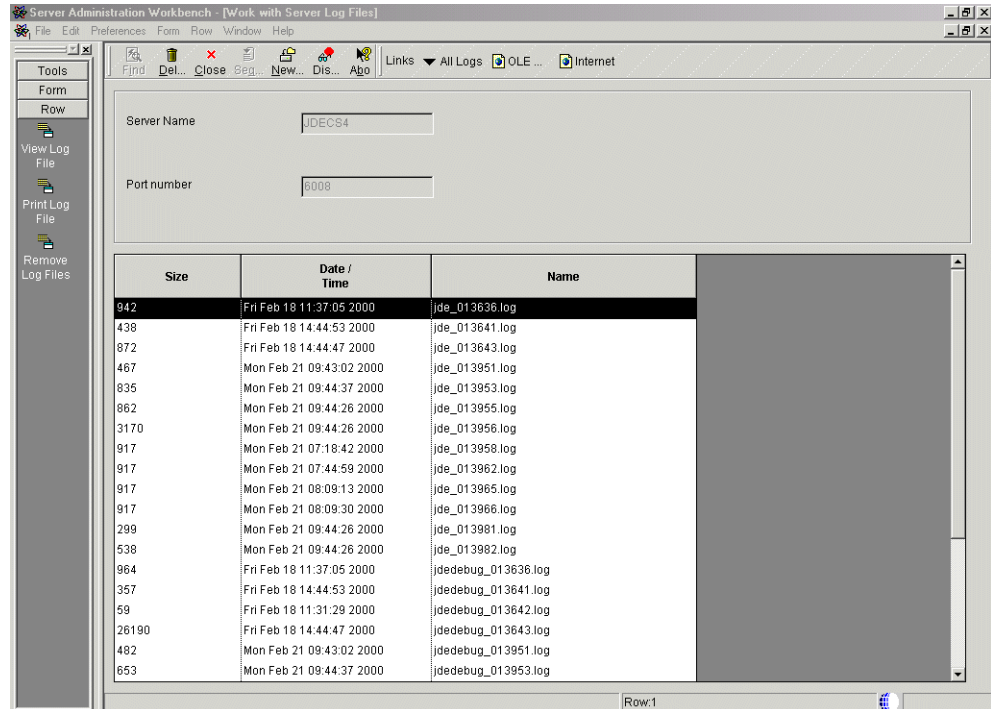


Figure 331. View log file

This opens the appropriate log file.

For more information, see “Server Administration Workbench (SAW)” in the *OneWorld Server and Workstation Administration Guide*.

21.3.9.7 J.D. Edwards PC tools

Two separate PC tools are available from J.D. Edwards to help analyze the table and API activity from the client. They both produce similar output. One tool (ODBCLOG) uses the ODBC trace as input. The other tool (DBLOG) uses the JDEDebug log as input.

Both of these tools were developed through Worldwide Advanced Technologies to assist in the interrogation and interpretation of the large amount of trace data produced in the respective input file. Because DBLOG and ODBCLOG assume that the entire log file represents the subject time period, the user must be careful to control the creation of the file and logging of events.

DBLOG and ODBCLOG provide a listing of the file activity during the captured processing period. The program counts the number of reads, writes, updates, and deletes that occur against each file. This is accomplished by searching the log records for SQL statements with the keywords: SELECT, FETCH, INSERT, UPDATE, and DELETE. DBLOG and ODBCLOG identify the file referenced and tally the appropriate type of access.

DBLOG also codes the accesses to “cached” data in the debug log by preceding the SQL access statement with the key word “RESET”. For example, RESET: SELECT indicates that a “read” was performed against a “cached” file. To

delineate these types of file accesses, the DBLOG output indicates a separate count of each RESET coded SQL statement.

DBLOG and ODBCLOG both present their analysis in a report with columns for the file name and the type of file access. Figure 332 shows an example of a report with the file names and number of Selects, Fetches, Updates, Inserts, and Deletes for all the SQL statements.

File Name	Selects	Fetches	Inserts	Updates	Deletes	Total
B731OBJ.F986!	1	0	0	0	0	1
F0082	10	18	0	0	0	14
F00821	50	337	0	0	0	70
F0083	92	82	0	0	0	92
F0084	142	139	0	0	0	142
jdesys.F0092	3	0	0	0	0	3
idemp.F00921	2	0	0	0	0	2

Figure 332. DBLOG analysis report showing all SQL statements

Figure 333 shows an example of the API call report.

API Name	Number of Calls
JDB_BeginTransaction	9
JDB_ClearSelection	98
JDB_ClearSequencing	95
JDB_CloseTable	92
JDB_CloseView	1
JDB_Fetch	1014
JDB_FetchKeyed	283
JDB_FetchMatchingKey	312

Figure 333. DBLOG report showing API calls

21.3.10 Network and communication analysis tools

Network tuning is key to achieving good performance. The network frequently can be the slowest link in the system. Typically, you can improve network performance by reducing the network traffic or by reducing the number of turns across the WAN network to a minimum.

Communications trace is an AS/400 communication analysis tool available from IBM. Another widely used communication analysis tool is called a *sniffer*.

21.3.10.1 AS/400 communications trace

The OS/400 communications trace facility is an important tool for isolating both performance problems and error conditions. It traces and formats any communications type that has a line description (for example, Token-Ring, Ethernet, and SDLC).

AS/400 communications traces are helpful in diagnosing where a performance delay is occurring. Use the timestamp and the eye-catcher fields to measure how long the AS/400 system takes to process a request.

Communications traces also capture the data stream being passed between Client Access ODBC and the AS/400 database host server. IBM Support can break down this data stream to see what is being passed between the PC and the AS/400 system, including the SQL conversation and SQL statements, data definitions, and errors.

Collecting communications trace

Complete the following steps to perform a communications trace data collection:

1. On the OS/400 command line, type the `STRSST` command. Your user profile should have all object authority.
2. Enter option 1, Start a Service Tool, and press the Enter key.
3. Enter option 3, Work with Communications Trace, and press the Enter key.
4. If the name of the line you want to trace is under the Configuration Object field, type 8 in the Opt field. Press the Enter key to access the Start Trace display. If the name is not there, press F6 to access the Start Trace field.

5. Type the following code:

```
Configuration object.....(line description name)
Trace description.....(Optional)
Buffer size.....(6 or 7, the maximum size available)
```

Leave the other fields with the default values.

6. Press the Enter key.

7. R410 and later systems prompt for Trace Options:

```
Configuration object.....SITETRN
Type .....LINE
```

Select one of the following options:

- All data (no filtering)
- Remote controller data
- Remote MAC address data
- Remote SAP data
- Local SAP data
- IP protocol number
- IP address data

If the trace will run for a long time, such as when working on intermittent problems, you should take the appropriate filtering option to reduce the amount of data collected. Select option 2 for SNA connections or option 7 for TCP/IP. Enter the controller name or IP address. Press the Enter key to return to the Work with Communications Traces display.

Note: The trace is now active.

8. Re-create the problem to be traced.

Ending the communications trace

To end the communications trace, from the Work with Communications Traces display, enter option 2, Stop trace. Press the Enter key.

Formatting and printing the communications trace report

From the Work with Communications Traces display, perform the following steps:

1. Enter option 6, Format and print trace, and press the Enter key.
2. Format the trace.

SNA connections are usually formatted as shown in the following example. You must know the controller name of the PC.

```
Controller.....(name of the PC controller)
Data representation.....2 1=ASCII, 2=EBCDIC, 3=*CALC
Format RR, RNR commands....N          Y=Yes, N=No
Format SNA data only.....Y           Y=Yes, N=No
```

TCP/IP and AnyNet connections are usually formatted as shown here. You need to know the IP address of the PC.

```
Data representation . . . . . 2          1=ASCII, 2=EBCDIC, 3=*CALC
Format Broadcast data . . . . . N          Y=Yes, N=No
Format TCP/IP data only . . . . . Y          Y=Yes, N=No
```

Type choices, and press Enter.

```
IP address . . . . . (The IP address for the PC)
IPX/SPX connections are usually formatted as follows.
Data representation . . . . . 2          1=ASCII, 2=EBCDIC, 3=*CALC
Format Broadcast data . . . . . N          Y=Yes, N=No
Format IPX data only . . . . . Y          Y=Yes, N=No
```

Leave the other fields with the default values.

3. Press the Enter key, and wait for the Format of trace data complete message. This produces a spooled file owned by the emulation session's job.
4. Press F3 three times to reach the Exit System Service Tools display.
5. Press the Enter key to end the STRSST command.
6. Type the WRKSPLF command, and press the F18 (Bottom) to locate the formatted traces.

The spooled files should have the file name QPCSMPRT, with User set to the user profile that ran the trace. Send the trace as requested by IBM Service.

21.3.10.2 Sniffer

A network sniffer is a non-invasive device that captures all the traffic on a communications facility. It is good at analyzing and diagnosing lower level communications protocols. For example, it allows you to determine whether there are hardware related problems or addressing problems, or to track the number of retransmissions at the transport layer. All of these issues can affect the performance across the network. It can also be used for helping to isolate performance issues within applications. Sniffers basically produce raw data that can be analyzed either by the sniffer itself or by another tool. The network software tools usually have some degree of sniffing capability built in.

21.3.10.3 Other network analysis tools

Optimal Networks is a one-to-one relationship network analysis tool that specializes at the application layer. This enables you to quickly visualize response time performance, client/server database access, communication threads, and to perform such functions as response time predictions under

various LAN and WAN topologies. For information about this product, visit the Web site: <http://www.optimal.com>

CACI is a one-to-many network analysis tools with in-depth information on communications performance. For information about this product, visit the Web site: <http://www.caci.com>

MIL3 is a network planning tool that offers modeling, simulation, and an analysis of the performance of communications networks, computer systems and applications, and distributed systems. The models fit together in a hierarchical fashion. For information about this product, visit the Web site: <http://www.mil3.com>

21.4 AS/400 performance data collection procedures

The procedures described in this section are intended to assist a OneWorld AS/400 customer in retrieval of key performance information on their system. The customer may be requested to run one or more of the following performance capture procedures, depending on the nature of the performance evaluation. Since some of these tools are resource intensive, it is important to consider the impact on the entire AS/400 user community when performing each capture. Some of these procedures require a dedicated environment. Whenever the test requires that OneWorld services be ended, we recommend that you coordinate this activity with all of your OneWorld users.

Creating a performance analysis work library

All performance evaluations require the creation of a work library on your AS/400 system to consolidate all performance-related information. Follow the procedure described in 21.4.1, "Packaging performance evaluation data" on page 670, to submit this library for analysis. The naming convention for this library should be "JDE", followed by your machine serial number, for example, JDE12345. You should suppress any special characters that are not supported by OS/400 library naming conventions. This work library will hereafter be referred to as "YourLibrary". Use the following commands:

```
DSPSYSVAL SYSVAL(QSRLNBR)
CRTLIB LIB(JDE12345) TEXT('JDE Performance analysis library')
```

Capturing AS/400 software release levels, PTFs, and AS/400 resources

All performance issues should be submitted with this base information. It provides information critical for determining your release of OS/400, PTF level, and global system values. This information can help identify potential performance improvements in OS/400 and identify key IBM PTFs which are known bug fixes for OneWorld functionality. Complete the following steps:

1. Display AS/400 software resources by running this command:

```
DSPSFWRSC OUTPUT(*OUTFILE) OUTFILE(YourLibrary/JDESFWRSC)
```

2. Display all installed AS/400 PTFs with this command:

```
DSPPTF OUTPUT(*OUTFILE) OUTFILE(YourLibrary/JDEPTFS)
```

3. Create a physical file in your performance analysis library to hold system values by entering this command:

```
CRTPF FILE(YourLibrary/JDESPLF) RCDLEN(132) MBR(*NONE)
MAXMBRS(*NOMAX)
```

4. Print all system values (do not log off until this procedure is completed):

```
WRKSYSVAL OUTPUT(*PRINT)
```

5. Copy your system values from the spooled file to your performance analysis library holding file:

```
CPYSPLF FILE(QSYSPRT) TOFILE(YourLibrary/JDESPLF) SPLNBR(*LAST)  
TOMBR(JDESYSVAL)
```

6. Create an output file of the available hardware resources:

```
DSPHDWRSC TYPE(*AHW) OUTPUT(*OUTFILE)  
OUTFILE(YourLibrary/JDEHWRSC) OUTFILEMT(*TYPE2)
```

Procedure 1: Server performance data collection

This section outlines the steps required to perform general One World server performance data collection on the AS/400 system. This data collection process gathers information on overall AS/400 system performance, and should run during peak business hours. Complete this process:

1. Start the Performance Monitor data collector:

```
STRPFRMON MBR(PFRTST#1) LIB(YourLibrary) INTERVAL(5) ENDTYPE(*NOMAX)  
TRACE(*NONE)
```

2. Start the Database Monitor data collector:

```
STRDBMON OUTFILE(YourLibrary/DBMON) OUTMBR(PFRTST#1) JOB(*ALL)  
TYPE(*SUMMARY)
```

Note: Do not use TYPE(*DETAIL) due to the overhead running this method.

3. Continue to use the AS/400 system in a normal business manner for one to two hours. We recommend that this test represent actual business transactions during a heavy load interval.

4. End the Database Monitor collector:

```
ENDDBMON JOB(*ALL) COMMENT('Description of Application Test')
```

5. End the Performance Monitor data collector:

```
ENDPFRMON
```

Procedure 2: Detailed application performance data collection

This section outlines the steps required to perform detailed OneWorld application performance data collection on the AS/400 system. These operations require that all production users to exit all One World applications because JDE network services are shut down during this process.

To minimize the impact on the production environment, this process is divided into two parts. The first part prepares the OneWorld software for detailed application performance data collection. The second part performs the actual data collection process.

1. Prepare OneWorld server software (Part 1)

Perform the following steps:

- a. End JDE services on the host by using these commands:

```
ENDNET  
CLRIPC
```

- b. Enable performance explorer data collection on One World programs.

Enter:

```
SBMJOB CMD(CHGSRVPGM SRVPGM(PRODB732/*ALL) OPTIMIZE(*FULL)
ENBPFCOL(*ALLPRC))
```

Note: This step can take several hours.

Restart JDE network services on the AS/400 system by using the command:

```
STRNET
```

2. Application performance data collection (Part 2)

Follow this process:

- a. End JDE services on the host by using these commands:

```
ENDNET
CLRIPC
```

- b. Create a performance explorer collection definition by using the following command statement:

```
ADDPXDFN DFN(PFRTST#1) TYPE(*STATS) JOB(*ALL) DTAORG(*FLAT)
```

- c. Increase the trace buffer size:

```
TRCINT SET(*SIZE) SIZE(*MAX)
```

- d. Start the Performance Monitor data collector:

```
STRPFRMON MBR(PFRTST#1) LIB(YourLibrary) INTERVAL(5) ENDTYPE(*NOMAX)
TRACE(*ALL)
```

- e. Start the Performance Explorer data collector:

```
STRPEX SSNID(PFRTST#1) OPTION(*NEW) DFN(PFRTST#1)
```

- f. Start the Database Monitor data collector:

```
STRDBMON OUTFILE(YourLibrary/DBMON) OUTMBR(PFRTST#1) JOB(*ALL)
TYPE(*SUMMARY)
```

- g. Start JDE services on the host:

```
STRNET
```

- h. Execute the One World applications that are experiencing performance problems.

- i. End JDE services on the host by using these commands:

```
ENDNET
CLRIPC
```

- j. End the Database Monitor collector:

```
ENDDBMON JOB(*ALL) COMMENT('Description of Application Test')
```

- k. End the Performance Explorer data collector:

```
ENDPEX SSNID(PFRTST#1) DTALIB(YourLibrary) TEXT('Description of
Application Test')
```

- l. End the Performance Monitor data collector:

```
ENDPFRMON DMPTRC(*YES)
```

- m. Remove the Performance Explorer definition:

```
RMVPEXDFN DFN(PFRTST#1)
```

- n. Restart One World network services:

```
STRNET
```

Procedure 3: OneWorld logging identification of performance issues

You perform this test when there is a specific UBE or application performance issue you want to submit for analysis. Reproduce the OneWorld application issue while OneWorld logging is activated. The content of the jde.log and jdedebug.log files assist in the analysis of OneWorld application and middleware issues. It is necessary to end OneWorld services during this test to activate full logging information. Be sure to deactivate full logging after this test is completed, since full logging has a negative impact on performance. Complete these tasks:

1. Activate the following jde.ini file settings on the Enterprise Server to perform full logging:

```
WRKMBRPDM B733SYS/INI
```

Enter a 2 to edit the JDE member.

```
[DEBUG]
Output=FILE
Trace=TRUE
LogErrors=1
KeepLogs=1

[JDEIPC]
ipcTrace=1

[JDENET]
netTrace=1

[UBE]
UBEDebugLevel=6
```

2. Delete all old IFS log files from your OneWorld IFS log directory. The location of your IFS log directories is found in the [DEBUG] section of the AS/400 jde.ini file. They are typically found in “/JDEB7321”.

```
RMVLNK OBJLNK('/JDEB7321/*.*')
```

3. Verify there are no active UBEs. End and restart OneWorld services.

Log on using a OneWorld user profile that has the library list: B733SYS, B732xSYS, B732xNET, QGPL and QCPA. Perform the following commands:

```
ENDNET
CLRIPC
STRNET
```

4. Recreate the application problem on your AS/400 system.
5. After recreating the application problem, capture the AS/400 jde.log and jdedebug.log from the IFS directory:

```
WRKLNK OBJ('/JDEB7321/*.*')
```

The log files have the job number appended to the end. You can use the OneWorld Work with Servers application to find the UBE job number to locate the correct JDE log and jdedebug.log stream file.

6. Create a multiple member Physical File to store your log files:

```
CRTPF FILE(YourLibrary/JDELOGS) RCDLEN(132) MBR(*NONE) MAXMBS(*NOMAX)
```

7. Copy the jde.log and the jdedebug.log file to the multi-member physical file just created:

```
CPYFRMSTMF FROMSTMF ('/JDEB7321/jde_063546.log')
TOMBR ('qsys.lib/YourLibrary.lib/jdelogs.file/jde_063546.mbr') MBROPT(*ADD)
```

```
CPYFRMSTMF FROMSTMF ('/JDEB7321/jdedebug_063546.log')
TOMBR ('qsys.lib/YourLibrary.lib/jdelogs.file/dbg_063546.mbr') MBROPT(*ADD)
```

8. Be sure to inactivate logging by changing the jde.ini file to the following settings:

```
WRKMBRPDM B733SYS/INI
```

Enter a 2 to edit the JDE member.

```
[DEBUG]
Output=NONE
Trace=FALSE
LogErrors=0
KeepLogs=0

[JDEIPC]
ipcTrace=0

[JDENET]
netTrace=0

[UBE]
UBEDebugLevel=0
```

9. End and Restart OneWorld services. Use the following commands:

```
ENDNET
CLRIPC
STRNET
```

Procedure 4: OneWorld journal file analysis

Journal files have significant runtime performance implications on the AS/400 system. It is desirable to identify all OneWorld files currently being journaled and create an outfile of results. This test helps identify if there are specific OneWorld application paper fixes, which may be applied to reduce the dependence of OneWorld journals. Follow this process:

1. Display your jde.ini file. Find the location of your OneWorld journal:

```
WRKMBRPDM B733SYS/INI
```

Enter a 2 to edit the JDE member.

```
[DB SYSTEM SETTINGS]
Default Journal=OW_JRNL
Default Journal Library=B733SYS
```

2. You should have already created a physical file. If not, create one now with the following command:

```
CRTPF FILE(YourLibrary/JDESPLF) RCDLEN(132) MBR(*NONE)
MAXMBRS(*NOMAX)
```

3. Print the files attached to your OneWorld journal:

```
WRKJRNA JRN(B733SYS/OW_JRNL) OUTPUT(*PRINT)
```

4. Convert your printed journal information to your performance analysis library file:

```
CPYSPLF FILE(QPDSPJNA) TOFILE(YourLibrary/JDESPLF) SPLNBR(*LAST)
TOMBR(JDEJRNA)
```

5. Identify all journal receivers to evaluate their size and ASP:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*JRNRCV) OUTPUT(*OUTFILE)
OUTFILE(YourLibrary/JDEJRNRCV)
```

Procedure 5: OneWorld table index analysis

The availability of table indexes has an effect on the behavior of OneWorld applications in general. It is valuable to record all OneWorld table indexes for the environment being evaluated. It is necessary to identify the library names of Business Data, Control Tables, and Object Librarian data sources used in your environment. If you are a co-existence customer, also include the COMMON library. You can identify these libraries by using OneWorld application “P986115 Data Sources”. Select the System data source, select each data source, and record the name in the Library Name field. These are the libraries you use for the following DSPDBR commands:

1. Collect Database Relationships. The first DSPDBR command creates an outfile. The remaining DSPDBR commands append to the existing outfile created by the first command:

```
DSPDBR FILE(Object Librarian Library/*ALL) OUTPUT(*OUTFILE)
OUTFILE(YourLibrary/JDELOGICAL)
```

```
OUTMBR(*FIRST *REPLACE)
```

```
DSPDBR FILE(Control Tables Library/*ALL) OUTPUT(*OUTFILE)
```

```
OUTFILE(YourLibrary/JDELOGICAL)
```

```
OUTMBR(*FIRST *ADD)
```

```
DSPDBR FILE(Business Data Library/*ALL) OUTPUT(*OUTFILE)
```

```
OUTFILE(YourLibrary/JDELOGICAL)
```

```
OUTMBR(*FIRST *ADD)
```

2. If you are a co-existence customer, also run the following command:

```
DSPDBR FILE(Common Library Name/*ALL) OUTPUT(*OUTFILE)
```

```
OUTFILE(YourLibrary/JDELOGICAL) OUTMBR(*FIRST *ADD)
```

3. Capture physical and logical file index attributes for key tables:

```
DSPFD FILE(Business Data Library/F09*) TYPE(*ATR) OUTPUT(*OUTFILE)
```

```
FILEATR(*PF) OUTFILE(YourLibrary/JDEINDEXPF)
```

```
DSPFD FILE(Business Data Library/F09*) TYPE(*ATR) OUTPUT(*OUTFILE)
```

```
FILEATR(*LF) OUTFILE(YourLibrary/JDEINDEXLF)
```

Procedure 6: Evaluation of the OneWorld CNC configuration

The configuration and setup of your OneWorld environment have an impact on performance especially for WAN users. To record this information, you need to duplicate your OneWorld jde.ini, OCM, and data source files into your performance analysis library.

Using the IBM Create Duplicate Object command, copy the following OneWorld configuration files:

```
CRTDUPOBJ OBJ(INI) FROMLIB(B733SYS) OBJTYPE(*FILE) TOLIB(YourLibrary)
DATA(*YES)
```

```
CRTDUPOBJ OBJ(F98611) FROMLIB(SYSB733) OBJTYPE(*FILE) TOLIB(YourLibrary)
DATA(*YES)
```



```
CRTDUPOBJ OBJ(F986101) FROMLIB(SYSB733) OBJTYPE(*FILE) TOLIB(YourLibrary)
DATA(*YES)
```

```
CRTDUPOBJ OBJ(F98611) FROMLIB(B733SYS) OBJTYPE(*FILE) TOLIB(YourLibrary)
NEWOBJ(F98611SM) DATA(*YES)
```

```
CRTDUPOBJ OBJ(F986101) FROMLIB(B733SYS) OBJTYPE(*FILE) TOLIB(YourLibrary)
NEWOBJ(F986101SM) DATA(*YES)
```

Procedure 7: SQL packages analysis

This data collection helps to analyze the setup and size of all SQL packages on your system. Create an output file of all SQL packages on your system in your work library:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*SQLPKG) DETAIL(*FULL) OUTPUT(*OUTFILE)
OUTFILE(YourLibrary/JDESQLPKG)
```

Procedure 8: Collecting the OneWorld Client jde.ini file and registry

From a functioning OneWorld client machine, capture the following files. Then, e-mail them to your J.D. Edwards performance specialist who is assigned to your account:

1. From the Windows or WINNT directory, copy the jde.ini file.
2. Export ODBC and Client Access registry entries:

- a. Click **Start**.
- b. Click **Run**.
- c. Perform regedit, and press Enter.
- d. Click **Registry->Export Registry File**.
- e. Click the selected branch, and enter:
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Client Access
- f. Save as File Name:
C:\registry.txt. Save as type: All Files(*.*)
- g. Click **Save**.
- h. Click **Registry->Export Registry File**.
- i. Click the selected branch, and enter:
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
- j. Save as File Name: ODBCsystemReg.txt
Save as type: All Files(*.*).
- k. Click **Save**.
- l. Click **Registry->Export Registry File**.
- m. Click the selected branch. Enter:
HKEY_CURRENT_USER\Software\ODBC\ODBC.INI
- n. Save as File Name: ODBCuser.txt
Save as type: All Files(*.*).
- o. Click **Save**.

21.4.1 Packaging performance evaluation data

Create a save file on your AS/400 system to package your performance analysis library. You can name your save file the same as your performance analysis library. Perform these steps:

1. Create a save file in the QGPL library:

```
CRTSAVF FILE(QGPL/YourLibraryName)
```

2. Save your performance analysis library to the save file you just created:

```
SAVLIB LIB(YourLibrary) DEV(*SAVF) SAVF(QGPL/YourLibraryName)
```

If you are sending this library on native tape format, substitute using the following command:

```
SAVLIB LIB(YourLibrary) DEV(TapeDeviceName)
```

You should receive a message stating *x* objects saved from library *YourLibrary*. The number of objects should match exactly the number of objects contained within this library. You can verify this by displaying the library for an object count:

```
DSPLIB LIB(YourLibrary)
```

21.5 OneWorld capacity planning with BEST/1

The objective of any capacity planning effort is to determine the optimum cost effective configuration required to support a specified workload (typically based on current, measured levels of activity), so acceptable response times are delivered to the user. This section presents an overview of the factors that you need to consider prior to launching a capacity planning project. These considerations are valid for most capacity planning efforts and are not confined to an AS/400 server environment.

Since it is not within the scope of this section to discuss performance tuning and performance optimization, we strongly recommend that you make every effort to make the application, its implementation, and its use as efficient as possible in the business environment. If this is not done as a prerequisite, any inefficiencies will be carried through with the increased workload and cause an increase in resource utilization.

21.5.1 Introduction

Capacity planning is a predictive process to determine future computing hardware resources required to support the estimated increases in computing workload. The increased workload can be a result of growth in business volumes or the introduction of new applications and functions to enhance the current suite of applications.

Due to its predictive nature, capacity planning can only be an approximation at best. The implementation of the same application in different environments can result in significantly different computing system requirements. There are many factors that can influence the degree of success in achieving the predicted results. These include changes in application design, the way users interact with the application, and the number of users who may use the applications. It is also difficult to determine external factors that affect the distribution of the workload over a given period of time, such as phone-in customer orders during a work day.

The objective of this section is to present techniques that assist in the prediction of AS/400 system requirements where the AS/400 functions as a server in a client/server computing environment. No attempt is made to determine PC client or network requirements.

The functions of the AS/400e capacity planning facility (BEST/1), available as part of the AS/400e Performance Tools Licensed Program (5769-PT1) Manager Feature, are used to predict AS/400 requirements. Detailed information on the use of this function is available in *BEST/1 Capacity Planning Tool*, SC41-3341.

This section introduces the key aspects of BEST/1 that need to be considered in capacity planning projects for AS/400 server applications. It is intended to help readers become familiar with the relevant BEST/1 facilities. The suggested approach assumes that performance data collected with the Performance Monitor is available to build the model. However, the AS/400 Performance Monitor does not record non-5250 transactions. Therefore, there is no granularity in the measurement of resource usage in a AS/400 server application. AS/400 performance measurements do not recognize individual requests to or responses from a server job. Also, all server applications are “batch-like”, and no LAN or WAN times are attributed to them. Therefore, it is not possible to use only the AS/400 measured performance data for capacity planning if you require some response time indications.

Any references to “transactions” and “response times” from reports produced by Performance Tools for AS/400 (5769-PT1) refer to interactive work produced by 5250 sessions and *do not apply* to any server applications that may be running at the time. The reports produced using the standard performance tools options provide you with information on the total use of resources such as processor, disk, and so on for server jobs.

There are many ways to implement a client/server application with the AS/400 functioning as a server. This section provides an overview of the method that applies to server workload modeling in general. Subsequent sections of this document present specific examples of the applicability of this process.

Note

This section presumes that you have at least a moderate level of understanding and experience with BEST/1. We recommend that you have access to the manual *BEST/1 Capacity Planning Tool*, SC41-3341, when reading this section.

The success of the approach covered in this section depends on your familiarity with the BEST/1 capacity planning interfaces and options as well as “artistic adjustments” by the BEST/1 user to define a non-interactive transaction within a BEST/1 workload.

You may have experience with client/server modeling techniques that are different than those illustrated in this section. If you have been successful using these techniques for doing capacity planning, we encourage you to continue using them. However, the techniques used in this section are also believed to be valid.

The keys to successful client/server capacity planning are a thorough understanding of the application implementation and use of collected performance data.

21.5.2 AS/400 server modeling

There is no intent to discuss designing applications for performance, performance management techniques, or system tuning options. The projections start from a known, measured workload and extrapolate the resource utilization and response times using a specified rate of growth. The basic prerequisites are that:

- The application is running optimally at expected levels of performance.
- There are no application-dependent constraints that invalidate BEST/1 extrapolations.
- No changes are expected in patterns of application usage or user behavior.
- A particular period of measured activity on the AS/400 system can be established as being representative of a sustained high peak workload.
- The proposed increase in workload can be related to the activity measured.
- An acceptable average response time estimate is established.
- A suitable capacity planning unit of measure has been determined based on application knowledge and measurability.

The key aspects covered in this section include:

- Model creation using measured data for a non-interactive workload
- Model validation
- Growth analysis

21.5.3 Assumptions

The capacity planning process makes the following assumptions:

- The applications under consideration have been optimized for the environment it is running in with regard to:

- Functions
 - Business requirements
 - User behavior
- The performance measurements used in the capacity planning project are a good representation of a typical busy workload on the system, including the mix of activity and volume of work.
 - There are no application dependent bottlenecks that prevent growth in throughput or improved response times. For example, an application may support only a single communications I/O Processor, or there may be a common code-path within the application that only allows requests to the Application Server to be single threaded.
 - The performance data is collected on a system that has not reached a saturation point with regard to any key system resources such as memory, processor, disk, and so on. When system resources are saturated, they introduce overheads, such as queuing, which are difficult to isolate and eliminate in the modeling process.
 - The nature of applications and its complexity do not change significantly as a part of the capacity projection. For example, a version upgrade of an application suite often results in significant changes in resource requirements associated with additional functions, processing architecture, and user behavior.
 - There is no change in the overall use of any functions of the application that result in increased usage of system resources.

If any of these assumptions are not valid, plan to make allowances for these increases in the projected workload increases.

21.5.4 Understanding the application environment

Consider the suite of applications whose workloads are to be extrapolated to provide for growth together with any satellite applications with which it is associated. An increased workload on one application may cause a cascading effect on related applications and increase their workload as well. For example, a planned increase in the number of users for a sales order entry system can result in increased activity in related systems such as materials management, production planning, accounting, and so on, as well as extend any batch run times such as periodic accounting statements to customers.

You also need a good understanding of the overall architecture of the application. For example, you need to know if the application uses a distributed logic approach with a significant amount of processing being performed at the client. Associated with this type of application, there may be a significant workload on the communications bandwidth resulting from high data volumes and line turnarounds. This knowledge can help you to determine the impact of the capacity of the AS/400 server on the total end-to-end client/server throughput and response. It can also assist you in setting levels of expectation with regard to the results that may be achieved once your recommendations are implemented.

A good understanding of the application can also assist in taking measurements that enable you to create a reasonably accurate workload model for extrapolation.

21.5.5 Selecting a capacity planning unit of measure

Identifying a measurable capacity planning unit of measure is critical to the success of a capacity planning project. The selected unit of measure must be clearly understood and have boundaries that can be defined and identified.

21.5.5.1 Measurability

It is important to measure a specific workload with regard to:

- Response time per Capacity Planning Units of Measure (CPUM)
- Number of CPUMs per unit of time
- Components included in the measure
- Overall AS/400 resource utilization such as:
 - CPU
 - Memory
 - Disk

The application provider may have included the capacity to measure this within the Application Server code. The Application Server code on the AS/400 system can recognize the start and end for processing a request from a client. The elapsed time between receiving the request and providing the response is the *response time* of the selected capacity planning unit of measure (CPUM). This gives you a value that represents the AS/400 server response time. At the same time, the server code can count the number of capacity planning units of measure (CPUMs) and report this at the end of a measured period.

In cases where the Application Server has to go to a Database Server for data (such as in a multi-tier implementation), the Application Server may include the database service time in the response time measured at the Application Server. If it includes the external Database Server time, you must consider the impact of any adverse performance on the Database Server and the communications facilities on the measured response time.

21.5.5.2 Options

In certain applications situations, the measurement facility at the AS/400 server may not be readily available. For example, in an application using standard AS/400 ODBC server code, there is no standard facility to recognize a request or response, measure the response time, and count the number of requests or responses in a given period of time.

User exits available through the server code may be used, or code in the client application can be used to count the defined CPUMs and measure the response times.

If the measurements are made at the client, it should be recognized that, in addition to the AS/400 server response time, the measured response time includes:

- Any delay or overhead at the client
- Communications transmission delays and line turnaround times

You may even use business transactions such as sales orders, invoices, customer inquiries, payment vouchers, and so on as a capacity planning unit of measure. The volume of transactions are easily calculated for a period of time. However, each business transaction is the product multiple interactions with the computer system with some amount of key-think time between each interaction. In this

situation, the capacity planning model must make allowances for these external delays.

21.5.6 Determining growth objectives

An effective capacity planning project requires an estimation of the proposed increase in workload with which the computing system must cope. Often the business managers predict increases in terms of sales revenue, profit projections, or other business measurements. These business measurements have to be translated into the impact on the computing workload.

You must work closely with the customer to establish these increases and gain agreement on the estimated increases. A simple increase in volume at one level of the business can have a cascading effect on the overall computing activity on the system. Ensure that you document the assumptions and the basis on which the increases in computing workload were determined. This helps in reconciling any differences that may be encountered when the changes actually occur.

21.5.7 Setting expectations

A clear understanding of the expected deliverable of the capacity planning project must be agreed upon. It is important to stress the predictive nature of the process, and the fact that there are many factors that can influence the degree of success in achieving the predicted results.

There are margins for error at every stage of the predictive process including:

- Estimation of workload at the business level
- Existence of application dependant bottlenecks
- Translation of the business workload into CPUMs
- Prediction of user behavior
- Determination of periods of peak activity

For these reasons, the conclusions and recommendation of a capacity planning project can only be an approximation.

The customer should also indicate expected average response times. It is normally the practice to specify response times requirements below a particular value (for example, under 1.5 seconds) for a specified percentage of the transactions (for example, 90% of the interactive transactions).

Averages

The BEST/1 modeling facilities use averages, and their predictions of utilization and response times are also averages.

21.5.8 Measuring workload and system resource utilization

This is a critical aspect of the capacity planning project because it forms the basis of extrapolation for the growth estimates provided by the customer. Consider the following points:

- The duration over which performance data is collected should provide representative information over a period of high activity. If the customer cannot identify a period of sustained peak activity, you may have to measure

performance data over a period of time to establish a suitable time frame from which to develop a capacity planning model.

- Select a Capacity Planning Unit of Measure (CPUM) that can assist you in modeling the growth. The selected CPUM must be measurable in terms of the quantity and the response time over the measured period.
- Identify the various components of response that contribute to the CPUM response time so that the necessary adjustments can be made when determining the AS/400 resource contributions.

Building a workload model for BEST/1 requires the following basic information:

- AS/400 resource utilization information
- Application workload measurements:
 - Number of Capacity Planning Units of Measure
 - Average response time per CPUM
 - AS/400 resource category used by the applications

The first step is to collect AS/400 performance data and application workload statistics for the selected period of system activity. In a client/server environment, the information may have to be collected from more than one source if all the necessary information is not available through the AS/400 performance data. The two sets of data to be used in building the model should be from the same period in time so that the information can be correlated.

21.5.9 Collecting AS/400 performance data

Use the standard OS/400 `STRPFRMON` command for performance data collection to start the Performance Monitor (Figure 334). This provides information on the AS/400 resource usage during the period. Select the data collection time interval to provide sufficient granularity to identify a representative dataset for modeling.

```

                                Start Performance Monitor (STRPFRMON)

Type choices, press Enter.

Member . . . . . *gen          Name, *GEN
Library . . . . . gsjde         Name
Text 'description' . . . . . measures J.D. Edwards One World application

Time interval (in minutes) . . . 5           5, 10, 15, 20, 25, 30, 35...
Stops data collection . . . . . *ELAPSED     *ELAPSED, *TIME, *NOMAX
Days from current day . . . . . 0           0-9
Hour . . . . . 1                0-999
Minutes . . . . . 0             0-99
Data type . . . . . *ALL         *ALL, *SYS
Select jobs . . . . . *ALL       *ALL, *ACTIVE
Trace type . . . . . *NONE       *NONE, *ALL
Dump the trace . . . . . *YES     *YES, *NO
Job trace interval . . . . . .5     .5 - 9.9 seconds
Job types . . . . . *DFT         *NONE, *DFT, *ASJ, *BCH...
                                + for more values
                                More...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
  
```

Figure 334. Start Performance Monitor (STRPFRMON) command entry

Enter the name of an existing library to collect performance data.

Enter an appropriate description for the dataset.

Enter a time interval for data “snapshots”. We recommend that you enter a value of 5 minutes.

Note

The duration over which performance data is collected can be specified at the time STRPFRMON is run or can be stopped by running the ENDPFRMON command.

21.5.10 Collecting OneWorld application workload data

The major components of the application environment that need to be captured are:

- Number of CPUs
- Average response time per CPU
- AS/400 resource allocation category used by the applications (such as job numbers, subsystem name, and so on)

The OneWorld application does not collect any specific information that can be used directly as a capacity planning unit of measure. However, there are some indicators stored in database files that can be used indirectly to draw conclusions on the amount of activity within different applications.

The control file for the “next number” in OneWorld can be used as an indicator. It contains a record for each major application code that can be executed and a set of counters associated with that application code. For example, if the application needs to add a customer record, an inventory item, etc., it needs to update and increment one of the counters in a particular record in the “next number” file (F0002) in PRODCTL.

Here is an example of the layout of this file:

Field	Data Type	Field Length	Buffer Length	Buffer Position	Field Usage	Column Heading
NNSY	CHAR	4	4	1	Both	NNSY
	Allows the null value					
	Coded Character Set Identifier :					
NNUD01	CHAR	15	15	5	Both	NNUD01
	Allows the null value					
	Coded Character Set Identifier :					
NNUD02	CHAR	15	15	20	Both	NNUD02
	Allows the null value					
	Coded Character Set Identifier :					
NNUD03	CHAR	15	15	35	Both	NNUD03
	Allows the null value					
	Coded Character Set Identifier :					
NNUD04	CHAR	15	15	50	Both	NNUD04
	Allows the null value					
	Coded Character Set Identifier :					
NNUD05	CHAR	15	15	65	Both	NNUD05
	Allows the null value					
	Coded Character Set Identifier :					
NNUD06	CHAR	15	15	80	Both	NNUD06
	Allows the null value					
	Coded Character Set Identifier :					
NNUD07	CHAR	15	15	95	Both	NNUD07
	Allows the null value					

	Coded Character Set Identifier	37		
NNUD08	CHAR 15 15 110	Both	NNUD08	
	Allows the null value			
	Coded Character Set Identifier	37		
NNUD09	CHAR 15 15 125	Both	NNUD09	
	Allows the null value			
	Coded Character Set Identifier	37		
NNUD10	CHAR 15 15 140	Both	NNUD10	
	Allows the null value			
	Coded Character Set Identifier	37		
NNN001	PACKED 8 0 5 155	Both	NNN001	
	Allows the null value			
NNN002	PACKED 8 0 5 160	Both	NNN002	
	Allows the null value			
NNN003	PACKED 8 0 5 165	Both	NNN003	
	Allows the null value			
NNN004	PACKED 8 0 5 170	Both	NNN004	
	Allows the null value			
NNN005	PACKED 8 0 5 175	Both	NNN005	
	Allows the null value			
NNN006	PACKED 8 0 5 180	Both	NNN006	
	Allows the null value			
NNN007	PACKED 8 0 5 185	Both	NNN007	
	Allows the null value			
NNN008	PACKED 8 0 5 190	Both	NNN008	
	Allows the null value			
NNN009	PACKED 8 0 5 195	Both	NNN009	
	Allows the null value			
NNN010	PACKED 8 0 5 200	Both	NNN010	
	Allows the null value			
NNCK01	CHAR 1 1 205	Both	NNCK01	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK02	CHAR 1 1 206	Both	NNCK02	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK03	CHAR 1 1 207	Both	NNCK03	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK04	CHAR 1 1 208	Both	NNCK04	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK05	CHAR 1 1 209	Both	NNCK05	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK06	CHAR 1 1 210	Both	NNCK06	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK07	CHAR 1 1 211	Both	NNCK07	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK08	CHAR 1 1 212	Both	NNCK08	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK09	CHAR 1 1 213	Both	NNCK09	
	Allows the null value			
	Coded Character Set Identifier	37		
NNCK10	CHAR 1 1 214	Both	NNCK10	
	Allows the null value			
	Coded Character Set Identifier	37		

The first field described in the above example, called NNSY, represents the system code, also called the application code. It can be used to identify the type of application being run (for example, 42, 48, 06, etc.).

The next ten fields (NNUD01 - NNUD10) are descriptions for up to ten transaction types within the application code (for example, Sales Order, Direct Ship, Sales Quote, etc.).

The next ten fields (NNU001 - NNU010) are actually the "next number" fields for the previously mentioned transaction types and potentially represent the activity being executed.

Note

For more information of the next number file (F0002 in library PRODCTL) and its exact interpretation of the contents for your particular installation, ask your J.D. Edwards representative.

The trick is now to take a snapshot of the contents of this file before the measurement begins and at the end of the measurement and calculate the delta.

For example, if you had 150 increments in the first record, an additional 250 in the second, and 600 in the third, you can assume that the application successfully ran 1,000 “transactions” or CPUMs per measurement.

If the measurement happened to be for one hour, you can directly use that value later on in modeling as an indicator for the amount of work being done. All the recorded resource utilizations can now be divided into these 1,000 CPUMs or now called OneWorld transactions.

It would be nice if this number would match exactly the number of customer transaction being executed, but it does not have to be the same. Actually, it can be anything. However, it is important that it is related to the activity being recorded during the measurement.

Using these CPUM values allows you to extrapolate an average response time for this activity. You may want to compare this with the average response time you measured with your stop watch at your client’s side. If the response time comes close, the number of assumed CPUMs are in the right range.

Important

Obtaining this information requires an understanding of the OneWorld application and the application codes stored inside the “next number” file. The response time is not collected anywhere, so you would have to use a stopwatch to measure response times!

The determined number of CPUMs or OneWorld application code counts are used later in this section to extrapolate the measured data.

21.5.11 Creating a model using measured data

BEST/1 provides the facility to use measured AS/400 performance data in building a model for capacity planning. The objective is to create separate *workloads* within the model to represent the various server applications that are involved in the capacity planning project.

Once the separate workloads are created, they can be reviewed for the effect of independent rates of growth on resource utilization and response time. When the workloads have been created, they can be saved and used in other BEST/1 models combined with other workloads.

The workloads can be defined using the following categories or groups:

- User ID
- Job type
- Job name
- Account code
- Job number
- Subsystem
- Memory pool

The steps for using BEST/1 to create a model with one or more distinct workloads based on your definitions are shown here:

1. Use the `STRBEST` command to activate the “advanced user level” of BEST/1.
2. Select the performance dataset to be used in modeling.
3. Identify the time period to be considered.
4. Classify the server jobs into suitable workloads.
5. Define the number of CPUMs per hour for each server workload.
6. Create the model.

The remainder of this section shows the steps in building a BEST/1 model and validating (calibrating) a model.

21.5.12 Creating a BEST/1 model

To create a BEST/1 model, perform following steps:

1. Sign on the system and start BEST/1 by typing the `STRBEST` command. Press F4, which prompts you for input (Figure 335).

```
Start BEST/1 (STRBEST)

Type choices, press Enter.

BEST/1 data library . . . . . jde      Name, *CURLIB
Performance data library . . . . . jde      Name
Log member . . . . . *NONE          Name, *NONE
Log library . . . . . *BESTDTAL     Name, *BESTDTAL
User level . . . . . *ADVANCED      *ADVANCED, *BASIC

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figure 335. Start BEST/1 (STRBEST) command entry

2. Enter the name of an existing library to contain BEST/1 objects created during modeling.

Enter the name of the library that has the previously collected performance data.

An initial menu display is shown after the disclaimer display is shown.

3. On the BEST/1 for the AS/400 menu, select option 1 (Work with BEST/1 models) as shown in Figure 336.

```
BEST/1 for the AS/400

Select one of the following:

    1. Work with BEST/1 models
    5. Create BEST/1 model from performance data
   10. Work with results
   50. General information and tutorial
   60. More BEST/1 options

Selection or command
====> 1

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F21=Basic user level
```

Figure 336. Select Work with BEST/1 models

Selecting option 1, Work with BEST/1 models, creates the display shown in Figure 337.

```
Work with BEST/1 Models

Library . . . . . GSJDE      Name
Type options, press Enter.
  1=Create  3=Copy  4=Delete  5=Work with  6=Print  7=Rename

Opt Model      Text      Date      Time
  1  JDEOW200

(No models in library)

Command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel
F15=Sort by model  F16=Sort by text  F19=Sort by date and time

Bottom
```

Figure 337. Work with BEST/1 Models

- Select option 1 to create a model, and enter a name for the model you want to build. The display shown in Figure 338 appears. Enter option 1, Create from performance data.

```

                                Create BEST/1 Model

Select one of the following:

    1. Create from performance data
    2. Create from predefined and user-defined workloads

Selection 1

F3=Exit  F12=Cancel

```

Figure 338. Create BEST/1 Model

- Enter the information shown in Figure 339, including the description text for your model, the member name containing the AS/400 performance data, and the name of the library containing the measured AS/400 performance data.

```

                                Create BEST/1 Model from Performance Data

Model . . . . . :  JDEOW200

Type choices, press Enter. Use *SLTHOUR to select an hour-long time period or
use *SLTTIV to select select first and last interval of a one to two hour
time period. The time period selected should be representative of your peak
processing activity.

Text . . . . . 200 user 1cpu 15MB/user local pkg cache

Performance data:
  Member . . . . . SOL01      Name, F4 for list
  Library . . . . . GSJDE     Name

Start time . . . . . *SLTTIV  Time, *FIRST, *SLTHOUR, *SLTTIV
Start date . . . . . *FIRST    Date, *FIRST

Stop time . . . . . *LAST     Time, *LAST
Stop date . . . . . *LAST     Date, *LAST

F3=Exit  F4=Prompt  F12=Cancel

```

Figure 339. Create BEST/1 Model from Performance Data

Note

Use the Start/Stop specifications to select the performance data you want to include in your model. Use the Help key for assistance on using these fields.

If you select *SLTITV in the display shown in Figure 339, a display that allows you to select the start time and end time of the performance data to be included in the model appears (Figure 340).

```

                                Select Time Interval
Library . . . . . :  GSJDE      Performance member . . :  SOL01
Type option, press Enter.  Select first and last interval.
  1=Select

```

Opt	Date	Time	---Transaction---		--CPU Util---		I/Os per Sec	
			Count	Rsp Time	Total	Inter	Sync	Async
	11/18/99	13:32:12	0	.0	1	0	143	7
	11/18/99	13:37:12	0	.0	4	3	140	16
1	11/18/99	13:42:12	0	.0	4	2	81	14
	11/18/99	13:47:13	0	.0	4	2	40	12
	11/18/99	13:52:15	0	.0	6	3	33	16
	11/18/99	13:57:15	0	.0	5	3	31	15
	11/18/99	14:02:15	0	.0	6	3	15	15
	11/18/99	14:07:16	0	.0	5	3	14	17
	11/18/99	14:12:17	0	.0	4	2	9	10
	11/18/99	14:17:18	0	.0	4	3	9	10
1	11/18/99	14:22:18	0	.0	4	2	10	9

More...

```

F3=Exit   F12=Cancel  F15=Sort by interval  F16=Sort by count
F17=Sort by rsp time  F18=Sort by total CPU util  F19=Sort by total I/Os

```

Figure 340. Select Time Interval for performance data

6. Select the start and end times only by entering a 1. Press Enter twice.
7. The display shown in Figure 341 on page 684 allows you to specify how you want to classify the resource usage into workloads. Enter option 2, and press Enter.

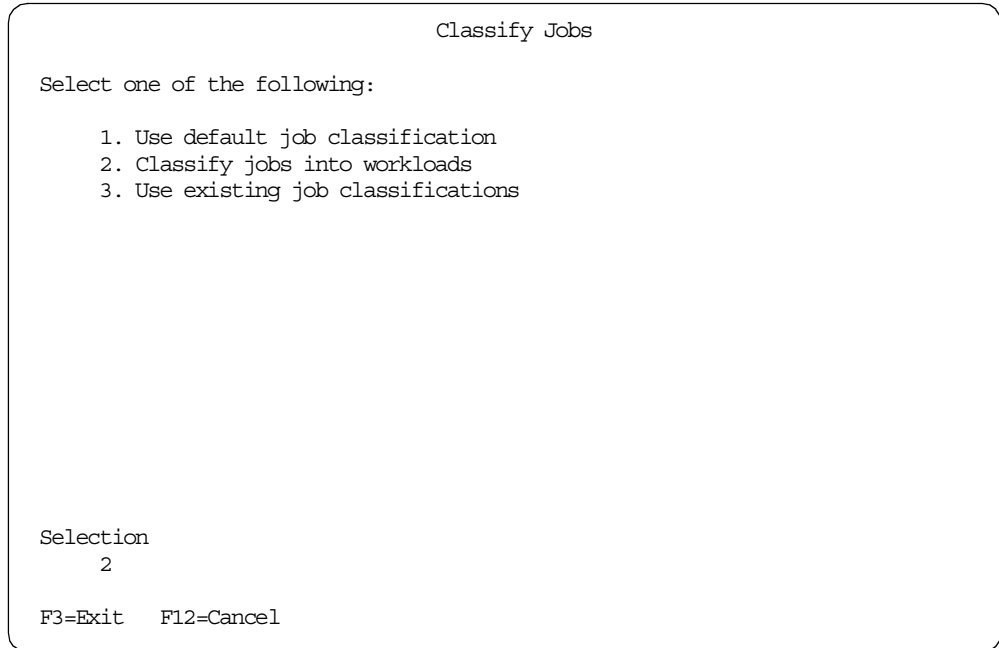


Figure 341. Classify Jobs

8. The display that appears (Figure 342) allows you to select the job category you want to use in assigning the AS/400 server jobs to workloads. Select the category that can assist you in defining your workloads. For example, if each of the server workloads run in separate subsystems, you may select option 6, Subsystem, as the classification category.

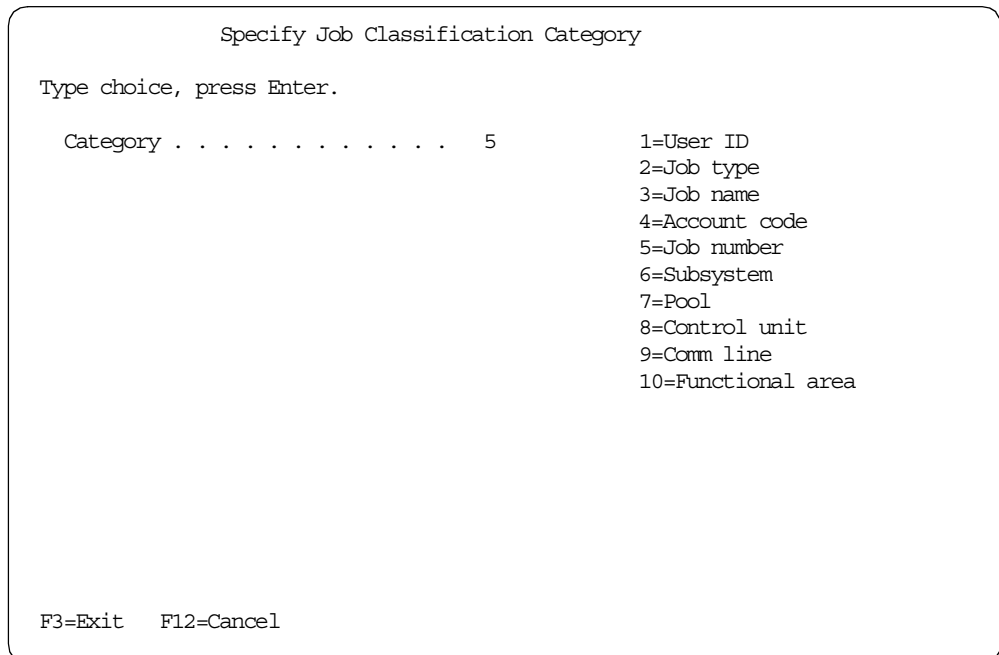


Figure 342. Specify Job Classification Category

9. Press Enter. The details of this display depend on your selection of the classification category. The example shown in Figure 343 is based on Job Number.

Edit Job Classifications

Enter workload names and category values which are assigned to each workload, press Enter. Jobs with unassigned values become part of workload QDEFAULT.

Workload	Job Name	Workload	Job Name	Workload	Job Name
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

More...

F3=Exit F9=Display values from data F12=Cancel
To display values from performance data, press F9

Figure 343. Edit Job Classification

Press F9 to analyze your performance data and present the available values within the category selected.

Assign the AS/400 server jobs to as many workloads as you want to create within your model. If your measured data has interactive work in addition to server applications, you can define these as well.

In Figure 344 on page 686, we show the panel where you assign specific jobs to a workload.

Note

We do not recommend explicitly assigning *LIC tasks to a specific workload. BEST/1 has internal algorithms to allocate *LIC tasks to the appropriate workload.

```

Assign Jobs to Workloads

Workload . . . . . JDE

Type options, press Enter. Unassigned jobs become part of workload QDEFAULT.
1=Assign to above workload 2=Unassign

Opt  Workload  Job Name      Number of      CPU           I/O
      Workload  Job Name      Transactions   Seconds       Count
1     1         IPRTR00049    0              9.926         8
      1         IPRTR00050    0              .000          0
      1         IPRTR00051    0              .000          0
      1         IPRTR00052    0              .000          0
      1         IPRTR00053    0              .000          0
1     1         IPRTR00054    0             13.582        12
1     1         QZDASOINIT    0            402.640      117268
      1         QZDASRVSD     0              .000          0
      1         QZHQSRVD      0              .000          0
      1         QZLSFILE      0              .000          0

More...

F3=Exit  F12=Cancel  F15=Sort by workload  F16=Sort by job name
F17=Sort by transactions  F18=Sort by CPU seconds  F19=Sort by I/O count

```

Figure 344. Assign Jobs to Workloads

10. Press Enter twice. On the panel shown in Figure 345, specify the type of paging behavior exhibited by the workload.

```

Specify Paging Behaviors

Type choices, press Enter.

Workload      Paging Behavior
              (F4 for list)
QDEFAULT      *GENERIC
JDE           *GENERIC

F3=Exit  F4=Prompt  F12=Cancel

Bottom

```

Figure 345. Specify Paging Behaviors

Accept the default value of *GENERIC for all workloads. Press Enter.

11. The display shown in Figure 346 allows you to specify the number of CPUMs for each server workload. Refer to 21.5.10, “Collecting OneWorld application workload data” on page 677, for a discussion on the selection of CPUMs.

CPUMs per hour

The value entered is *per hour* regardless of the time period over which the model was built.

Ensure that you change the Type field to *NONE for all the server workloads.

```
Define Non-Interactive Transactions
Job classification category . . . . . : Job Name

Type choices, press Enter.

Workload      ---Activity Counted as Transaction---      Total Transactions
QDEFAULT      Type          Quantity          when Type = *NONE
JDE           *LGLIO         100.0              0
              *NONE         100.0              2400

Type: *LGLIO, *CMNIO, *CPUSEC, *PRINT, *NONE
F3=Exit  F12=Cancel

Bottom
```

Figure 346. Define Non-Interactive Workloads

12. Press Enter. The display shown in Figure 347 on page 688 allows you to save the job classifications as a member in a library.

```

Save Job Classification Member

Change values if desired, press Enter.

Member . . . . . JDEOWTRANS      Name
Library . . . . . GSJDE          Name

Text . . . . . JDE OneWorld transactipons 2400/hour

Replace . . . . . N              Y=Yes, N=No

F12=Cancel

```

Figure 347. Save Job Classification Member

Enter the member name to contain the classification.

Check the library name for BEST/1 objects.

Enter description text for your classification.

13. Press Enter. Check the model and library name and the associated text, and press Enter to submit a batch job to create your BEST/1 model. You return to the Work with BEST/1 Models display (Figure 348).

```

Confirm Creation of BEST/1 Model

Type choices, press Enter.

Model . . . . . JDEOW200      Name
Library . . . . . GJSJDE      Name

Text . . . . . 200 user 1cpu 15MB/user local pkg cache

Replace . . . . . N          Y=Yes, N=No

Job name . . . . . CRTBESTMDL  Name, *JOB
Job description . . . . . QDFTJOB . . . . .Name, *NONE, *USRPRF
Library . . . . . QPFR        Name, *LIBL, *CURLIB

F12=Cancel
Member JDEOWTRANS has been saved

```

Figure 348. Confirm Creation of BEST/1 Model

Check the member name to contain the classification.

Check the library name for BEST/1 objects.

Check the description text for your classification.

You may issue the Work with Submitted Job (WRKSBMJOB) command to find out when your create model job completes, or issue the DSPMSG command and look for a model completion message. Or, you can press the F5 key (Refresh) repeatedly until you see your new model name appear on the Work with BEST/1 Models menu. After you see your new model on this display, proceed to the next step.

14. Validate the BEST/1 model.

Prior to evaluating the effect of workload growth, review the model created by BEST/1 to ensure that it conforms to the measured workload and response times.

Model objectives

A BEST/1 model may be considered to be properly calibrated when the measured data values have response times within .5 seconds of the corresponding predicted data values, and the resource utilizations between the measured and predicted are within 20% of each other. In server capacity planning, you can ignore any differences in communications resource utilization because you limit the modeling to the server only.

Note

The BEST/1 manual, *BEST/1 Capacity Planning Tool*, SC41-5341, discusses the adjustments required when modeling “server jobs”. You have to rely on your experience with BEST/1 and your understanding of the application and the OS/400 operating environment to effectively adjust or *calibrate* the model.

21.5.13 BEST/1 workload components

You may have to verify and modify the workloads if the analysis shows that BEST/1's calculations for some important resource usages do not coincide with actual measurements. These include:

- Total CPU utilization
- Disk I/O activity
- Number of non-interactive transactions per hour=Yes, N=No

21.5.13.1 Workload

A BEST/1 model is made of one or more user-defined *workloads*, which is the main unit of input to capacity planning.

From the Work with BEST/1 Model display, select option **1**, Work with workloads, to see all the workloads in a BEST/1 model. The display shown in Figure 349 appears.

```
Work with Workloads

Model/Text:  JDEOW200  200 user 1cpu 15MB/user local pkg cache

Type options, press Enter.
 1=Create  2=Change  3=Copy  4=Delete  5=Display  6=Print  7=Rename
 8=Save workload to workload member  9=Edit transactions

Opt  Workload  Text
-----
 5   JDE       Measured from GSJDE (SOL01)
     QDEFAULT  Measured from GSJDE (SOL01)

Bottom
F3=Exit  F6=Add saved workload  F9=Add predefined workload  F12=Cancel
F13=Combine workloads
```

Figure 349. Work with Workloads

21.5.13.2 Function

Each workload is made up of one or more *functions*. Each defined workload created using performance data defaults to one function per user per hour. This function represents all the work done by this workload. Enter option 5 to display the functions in a workload (Figure 350).

Note

Non-interactive workloads have a usage mode represented by "N/A".

```
                                Display Workload

Workload . . . . . : JDE
Workload text . . . . . : Measured from GSJDE (SOL01)
Workload type . . . . . : *NORMAL
CPU architecture . . . . . : *RISC
Usage mode . . . . . : 4          1=Casual, 2=Interrupted, 3=Steady,
                                4=N/A

Type option, press Enter.
 5=Display

Opt  Function  Function Text          Functions  Avg K/T    Inter Tns
   5  JDE      Function of JDE        per User   (secs)    per Function
                                1.00      N/A        .00

                                                                Bottom
F3=Exit  F9=Display chars to comm line resources  F10=Display I/Os to ASPs
F11=Display non-inter transactions  F12=Cancel
```

Figure 350. Display Workload

Two additional values are available at the Function level to assist you in calibrating a BEST/1 model. These are:

- Key/think time (not applicable for non-interactive work)
- Additional delays

These parameters can be modified by selecting the Change Function option as shown in Figure 351 on page 692.

```

                                Display Function
Workload . . . . . : JDE           Measured from GSJDE (SOL01)
Function . . . . . : JDE           Function of JDE

Key/Think time . . . . . : N/A           Seconds
Additional delays . . . . . : .0           Seconds

Type options, press Enter.
  5=Display

      Transaction  Pool      Transactions      CPU Time      Total
Opt   Type        ID    Priority  per Function   (Secs)        I/Os
  2   2           1      0         1.00         6733.683     43328.5
  2   2           2     20        2400.00      21.164       48.8

                                                    Bottom

Transaction Type: 1=Interactive, 2=Non-interactive

F3=Exit  F12=Cancel

```

Figure 351. Display/change Function

21.5.14 Transaction

Each function is made up of one or more *transactions*. Details of a function and the transactions included in the function can be displayed by selecting option 5. The transactions in a server workload are non-interactive. The display in Figure 352 shows the transactions per hour.

```

                                Display Function
Workload . . . . . : JDE           Measured from GSJDE (SOL01)
Function . . . . . : JDE           Function of JDE

Key/Think time . . . . . : N/A           Seconds
Additional delays . . . . . : .0           Seconds

Type options, press Enter.
  5=Display

      Transaction  Pool      Transactions      CPU Time      Total
Opt   Type        ID    Priority  per Function   (Secs)        I/Os
  2   2           1      0         1.00         6733.683     43328.5
  2   2           2     20        2400.00      21.164       48.8

                                                    Bottom

Transaction Type: 1=Interactive, 2=Non-interactive

F3=Exit  F12=Cancel

```

Figure 352. Display Function

The display in Figure 353 shows an example of the information included in a transaction, which is obtained by selecting option 5.

```

                                Display Transaction
Workload . . . . . : JDE           Measured from GSJDE (SOL01)
Function . . . . . : JDE           Function of JDE

Transaction Type . . . . . : 2           1=Inter, 2=Non
Pool ID . . . . . : 2
CPU Priority . . . . . : 20
CPU time . . . . . : 21.164           Secs (on B10)
Permanent writes . . . . . : 95.5           Percent
Chars transferred in . . . . . : 0
Chars transferred out . . . . . : 0
Exceptional wait . . . . . : .0           Msec
Paging behavior . . . . . : *GENERIC

                                Reads      Writes
Sync DB I/Os . . . . . : 3.4           .1
Async DB I/Os . . . . . : 9.6           7.3
Sync non-DB I/Os . . . . . : 28.2           .1
Async non-DB I/Os . . . . . : .1           .0
Press Enter to continue.

F3=Exit  F12=Cancel  F13=Display paging behavior

```

Figure 353. Display function details

Adjustments

You can make adjustments to the model at any or all of the following components:

- Workload
- Function
- Transaction resources F1

21.5.15 Working with BEST/1 models and results

On the Work with BEST/1 Models menu (Figure 354 on page 694), enter option 5, Work with, for your model.

```

Work with BEST/1 Models

Library . . . . . GSJDE      Name

Type options, press Enter.
  1=Create  3=Copy  4=Delete  5=Work with  6=Print  7=Rename

Opt Model      Text                                Date      Time

  5  JDEOW200   200 user 1cpu 15MB/user local pkg cache  02/24/00  14:31:23

Command
====>
F3=Exit   F4=Prompt  F5=Refresh      F9=Retrieve  F12=Cancel
F15=Sort by model  F16=Sort by text  F19=Sort by date and time

Bottom

```

Figure 354. Work with BEST/1 Models

Press Enter. On the Work with BEST/1 Model menu, select option 5, Analyze current model, as shown in Figure 355. This option runs the specified workload against the current configuration in the model.

```

Work with BEST/1 Model

Performance data . . . . : GSJDE (SOL01)
Model/Text . . . . . : JDEOW200 200 user 1cpu 15MB/user local pkg cache

Select one of the following:

  1. Work with workloads
  2. Specify objectives and active jobs

  5. Analyze current model
  6. Analyze current model and give recommendations
  7. Specify workload growth and analyze model

 10. Configuration menu
 11. Work with results

Selection or command
====> 5
F3=Exit   F4=Prompt  F9=Retrieve  F12=Cancel  F15=Save current model
F17=Analyze using ANZBESTMDL  F22=Calibrate model  F24=More keys
Model JDEOW200 has been read

More...

```

Figure 355. Work with BEST/1 Model options

Press Enter. Use the facilities within this function to evaluate the suitability of the BEST/1 model. If the values predicted by the model are substantially different from the values measured by the AS/400 Performance Monitor and the

application profile measurements, you must make the necessary “artistic adjustments” to the model.

Figure 356 shows an example of the screen you use to work with the results of the analysis.

Calibration expertise

This is where skill and experience play a major role.

```
Work with Results

Printed report text . . . . . 200 user 1cpu 15MB/user local pkg cache

Type options, press Enter.
 5=Display  6=Print

Opt   Report Name
      Measured and Predicted Comparison
      Analysis Summary
      Recommendations
      Workload Report
      ASP and Disk Arm Report
      Disk Resources Report
      Main Storage Pool Report
      Communications Resources Report
      All of the above

                                           Bottom
F3=Exit  F12=Cancel  F14=Select saved results  F15=Save current results
F18=Graph current results  F19=Append saved results  F24=More keys
Model saturated due to main storage pool constraints  +
```

Figure 356. Work with Results

Some of the possible anomalies are discussed in the following section.

21.5.15.1 Measured and predicted comparison

On the Work with Results menu, enter option 5 next to Measured and Predicted Comparison. This shows a column of measured statistics compared to the values evaluated by the BEST/1 model.

The example in Figure 357 on page 696 shows a relatively low usage of CPU (less than 5%).

Measured and Predicted Comparison		
	Measured	Predicted
Total CPU util :	4.3	4.3
Disk IOP util :	2.1	2.1
Disk arm util :	.8	.8
Disk IOs per second :	45.2	45.2
Multifunction IOP util :	2.3	1.4
Disk IOA util :	3.6	3.6
LAN IOA util :		.0
WAN IOA util :		.0
Integrated PC Server IOA util :	.0	.0
		More...
Interactive:		
CPU util :	.0	.0
Int rsp time (seconds) :	.0	.0
Transactions per hour :	2	0
Non-interactive thruput :	2471	2482
Performance estimates -- Press help to see disclaimer.		
F3=Exit F6=Print F9=Work with spooled files F12=Cancel		

Figure 357. Comparing measured and predicted values

21.5.15.2 Transactions per function

Ensure that the number of transactions per function match the value input to the model.

Adjusting transactions

BEST/1 always takes some of the unassigned OS/400 system work (pool 2) and Licensed Internal Code (LIC) work (pool 1) and “adds” a portion to all BEST/1 workloads because some system and LIC work was required to perform such tasks as manage the job and perform disk and communications I/O.

If the workload is made up of non-interactive jobs, but they are *not* classified as *BATCHJOB, the number of active jobs is set to 1. For example, you may have had six active jobs, but the workload was really derived from only four.

You may have to examine the number of active jobs shown in Figure 358 by entering option 2 from the Work with BEST/1 Model menu.

```

Specify Objectives and Active Jobs

Model/Text:  JDEOW200   200 user 1cpu 15MB/user local pkg cache

Type changes, press Enter.

Workload      Connect  Workload  Active  ----Interactive-----  Non-inter
Workload      Type     Type      Jobs    Rsp Time    Thruput    Thruput
JDE           *LOCAL  *NORMAL   1.0     .0           0           0
QDEFAULT     *LOCAL  *NORMAL   .1      .0           0           0
QDEFAULT     *LAN    *NORMAL   .1      .0           0           0

Bottom
F3=Exit  F11=Show all quantities  F12=Cancel  F15=Sort by connect type
F19=Work with workloads

```

Figure 358. Specify Objectives and Active Jobs

This difference can result in erroneous reporting of the number of non-interactive transactions for the function.

21.5.15.3 Manual calibration of resource usage

BEST/1 allows you to manually calibrate resource utilization, such as CPU and disk, by adjusting workload functions until the predicted utilizations are a close approximation to the measured values. Manual calibration requires a good understanding of the parameters you are changing.

Model calibration

Use F22 to enter manual calibration mode. While you are in manual calibration mode, any changes to the number of active jobs, or functions per user, do not affect total workload throughput ("transactions" per hour). This is because values for transactions per function are modified to offset changes to either active jobs or functions per user. In addition, all paging coefficients are recalculated during analysis.

After manual calibration, return to the Work with BEST/1 Model menu, and press F22 to exit the manual calibration mode.

21.5.15.4 Calibrating response times

Server applications are reported as non-interactive jobs, and any external interfaces are not recognized. Therefore, BEST/1 does not recognize the impact of any communications delays or overhead.

However, if you need to represent an "interactive response time" to a server application (because the user perceives performance from this viewpoint), you

can adjust BEST/1's representation of response time by adjusting the Additional delays value in the workload specification.

The response time indication visible in the Display Workload Report display should be considered as representing internal response time.

For example, on the Work with Results menu, enter option 5 to display the Workload Report. You can then display response times details by pressing F11. Add an amount equal to the difference between the perceived user response time and the internal response time shown in the Workload Report (Figure 359).

Display Workload Report							
Period: Analysis		CPU	Thruput	-----Response Times (Secs)-----			
Workload	Type	Util	per Hour	Internal	Local	LAN	WAN
JDE	2	3.2	2400	.5	.5	.0	.0
QDEFAULT	2	1.2	60	2.2	2.2	2.2	.0

Bottom

Type: 1=Interactive, 2=Non-interactive, 3=*BATCHJOB
 Performance estimates -- Press help to see disclaimer.
 F3=Exit F10=Re-analyze F11=Response time detail F12=Cancel
 F15=Configuration menu F17=Analyze multiple points F24=More keys

Figure 359. Display Workload Report

Go to the Work with BEST/1 Model menu, and select option 1, Work with workloads, followed by option 2, Change, for the workload. Then press F6 (Work with functions), select option 2, Change, for the workload, and increase the additional delay. See Figure 360.

```

Change Function

Workload . . . . . : JDE          Measured from GSJDE (SOL01)
Function . . . . . : JDE

Change fields, press Enter.
Function text . . . . . :          Function of JDE
Key/Think time . . . . . : N/A          Seconds
Additional delays . . . . . : .1          Seconds

Transaction  Pool          Transactions      CPU Time      Total
Type        ID      Priority      per Function  (Secs)        I/Os
  2          1          0              1.00        6733.683      43328.5
  2          2          20             2400.00        21.164        48.8

Transaction Type: 1=Interactive, 2=Non-interactive

F3=Exit  F6=Work with transactions  F12=Cancel

Bottom

```

Figure 360. Change Function

Exceptional wait time can also be added at the transaction level of the workload (Figure 361).

```

Change Transaction

Workload . . . . . : JDE          Measured from GSJDE (SOL01)
Function . . . . . : JDE          Function of JDE

Type choices, press Enter.
Transaction Type . . . . . : 2          1=Inter, 2=Non
Pool ID . . . . . : 2          F4 for list
CPU Priority . . . . . : 20
CPU time . . . . . : 21.164      Secs (on B10)
Permanent writes . . . . . : 95.5    0-100 percent
Chars transferred in . . . . . : 0
Chars transferred out . . . . . : 0
Exceptional wait . . . . . : .1          Msec
Paging behavior . . . . . : *GENERIC  F4 for list

-----Database-----      ----Non-database----
Reads      Writes      Reads      Writes
Sync I/Os . . . . . : 3.4      .1      28.2      .1
Async I/Os . . . . . : 9.6      7.3      .1      .0

F3=Exit  F4=Prompt  F6=Calculate B10 CPU time  F12=Cancel
F13=Change paging behavior

```

Figure 361. Change Transaction

21.5.16 Saving the BEST/1 model

When calibration of the BEST/1 model is complete, save the model and workloads so that you can use it later for modeling growth of this workload and for

modeling it with other measured data workloads. This is especially important if considerable manual adjustments are made to the model.

1. Go back to the Work with BEST/1 Model menu, and select option **1**, Work with Workloads.
2. Select option **8**, Save workload to workload member, for each of the workloads and save the workload.

Save Workload to Workload Member

Change values if desired, press Enter.

Member	JDE	Name
Library	GSJDE	Name
Text	Measured from GSJDE (SOL01)	
Replace	Y	Y=Yes, N=No
CPU architecture	*RISC	*CISC, *RISC

F12=Cancel

Figure 362. Save Workload

3. Enter a member name to contain the workload.
4. Enter the library name for BEST/1 objects.
5. Enter a description text for your workload.

The workload member is saved to a file called QACYWKLS in your library. You can use this workload in any other system, or you can add this workload to any other model from any other system.

6. On the Work with BEST/1 Model menu, press F15 (Save the current model), enter a name (Member) of the model, and press Enter to save the model. See Figure 363.


```

                                Save Current Model

Change values if desired, press Enter.

Save to Model member:
Member . . . . . JDEOW200      Name
Library . . . . . GSJDE        Name
Text . . . . . 200 user 1cpu 15MB/user local pkg cache
Replace . . . . . Y            Y=Yes, N=No

Externally described member information:

Save . . . . . N              Y=Yes, N=No
Member . . . . . *MEMBER      Name, *MEMBER
Library . . . . . *LIB        Name, *LIB
Text . . . . . 200 user 1cpu 15MB/user local pkg cache
Replace . . . . . *REP        Y=Yes, N=No, *REP

F12=Cancel

```

Figure 363. Save model

7. Enter a member name to contain the model.
8. Enter the library name for BEST/1 object.
9. Enter a description text for your model.

Generating an “Externally described member” is for experienced BEST/1 users who want to save the model in a format that can be downloaded to a personal computer for later processing by user-written programs.

Discussing these capabilities is beyond the scope of this redbook. Therefore, we select N for the Save option.

21.5.17 Planning for growth

You can now make simulations of the effect of various hardware configurations on a selection of growth rates by typing your options into the model. Note that each of the individual workloads that you defined in the model can have separate growth rates.

Note

The growth projections presented by the BEST/1 model evaluate only the capacity of AS/400 configuration. Some of the benefits of the operating system enhancements are also included on average. However, the model does not take into consideration any specific operating system codepath advantages or disadvantages that the application may encounter.

BEST/1 also cannot predict if the extrapolation of the measured workload results in growth beyond any functional or operational limitations of the application that may exist. These have to be managed separately.

Perform following steps:

1. From the main BEST/1 menu, select option 1 to display the Work with BEST/1 Models menu. Select option 1 on the menu shown in Figure 364 to display the models in the selected library.

```
Work with BEST/1 Models

Library . . . . . GSJDE      Name

Type options, press Enter.
  1=Create  3=Copy  4=Delete  5=Work with  6=Print  7=Rename

Opt Model      Text
-----
  5  JDEOW200   200 user 1cpu 15MB/user local pkg cache  02/24/00  16:47:43

Command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel
F15=Sort by model  F16=Sort by text  F19=Sort by date and time

Bottom
```

Figure 364. Work with BEST/1 Models

2. Press Enter. Select option 7 to specify the workload growth amounts and the number of periods you want to model growth. See Figure 365.

```
Work with BEST/1 Model

Performance data . . . . : GSJDE (SOL01)
Model/Text . . . . . : JDEOW200  200 user 1cpu 15MB/user local pkg cache

Select one of the following:

  1. Work with workloads
  2. Specify objectives and active jobs

  5. Analyze current model
  6. Analyze current model and give recommendations
  7. Specify workload growth and analyze model

 10. Configuration menu
 11. Work with results

More...

Selection or command
====> 7
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F15=Save current model
F17=Analyze using ANZBESTMDL  F22=Calibrate model  F24=More keys
Model JDEOW200 has been read
```

Figure 365. Work with BEST/1 Model

- Press Enter. The display shown in Figure 366 allows you to grow all the workloads by a single percentage for up to 10 periods.

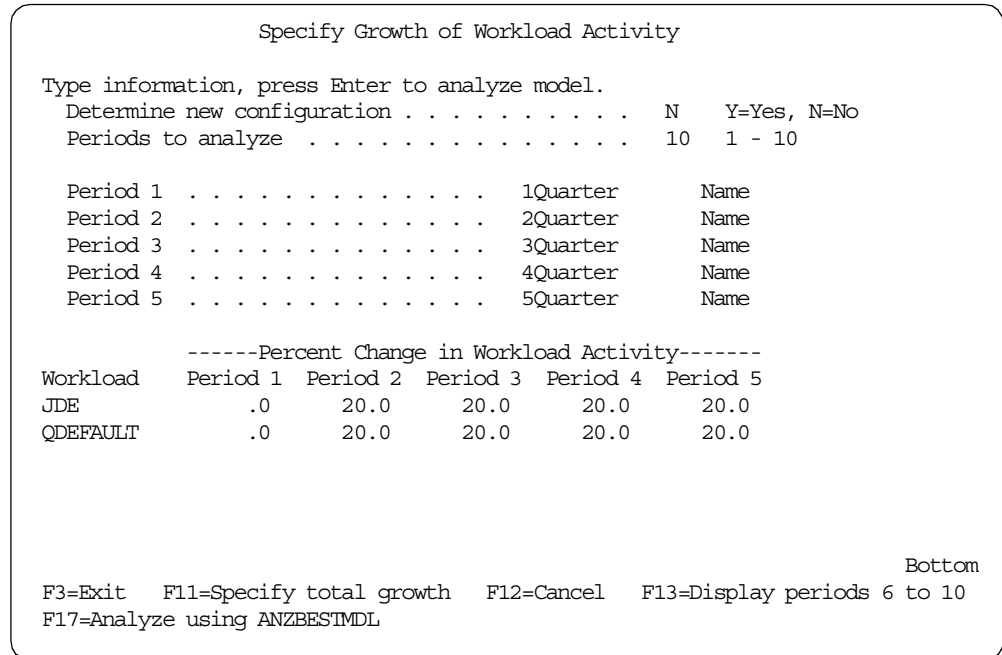


Figure 366. Specify Growth of Workload Activity

- Press F11 to display the workloads in the model that allows you to specify growth rates for each workload (Figure 367).

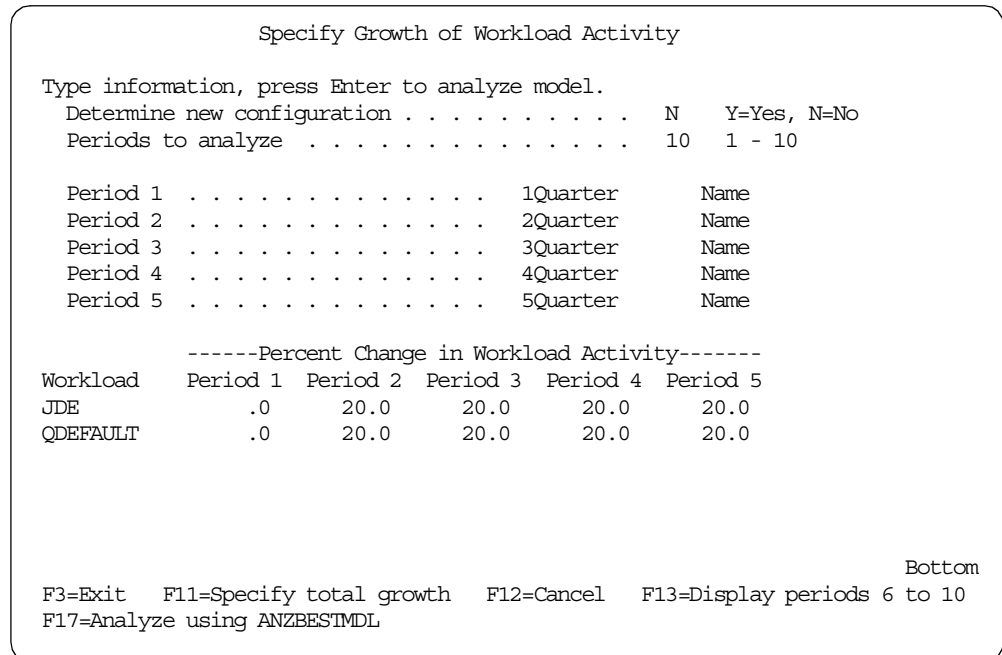


Figure 367. Specify Growth of Workload Activity: Detail

You can either have BEST/1 determine the required upgrade system, or you can modify the configuration and review the effect on system resource usage as the workload increases.

Status messages are issued as growth analysis is performed. When analysis is complete, the display shown in Figure 368 appears. Then, by entering option 5 next to *Analysis Summary*, the displays shown in Figure 369 through Figure 370 appear.

```
Work with Results

Printed report text . . . . . 200 user 1cpu 15MB/user local pkg cache

Type options, press Enter.
 5=Display  6=Print

Opt   Report Name
      Measured and Predicted Comparison
  5   Analysis Summary
      Recommendations
      Workload Report
      ASP and Disk Arm Report
      Disk Resources Report
      Main Storage Pool Report
      Communications Resources Report
      All of the above

Bottom
F3=Exit  F12=Cancel  F14=Select saved results  F15=Save current results
F18=Graph current results  F19=Append saved results  F24=More keys
Model has been analyzed
```

Figure 368. Work with Results

Display Analysis Summary									
Period	CPU Model	Stor (MB)	CPU Util	-Disk Nbr	IOPs-- Util	-Disk Ctls-- Nbr	Util	-Disk Arms-- Nbr	Util
1Quarter	730 2C6B	16384	4.3	3	2.1	28	.1	40	.8
2Quarter	730 2C6B	16384	5.2	3	2.5	28	.1	40	1.0
3Quarter	730 2C6B	16384	6.2	3	3.0	28	.1	40	1.2
4Quarter	730 2C6B	16384	7.5	3	3.6	28	.1	40	1.4
5Quarter	730 2C6B	16384	9.0	3	4.3	28	.1	40	1.7
6Quarter	730 2C6B	16384	10.8	3	5.2	28	.2	40	2.0
More...									
Period	--Non-Inter Rsp Time--			-----Non-Inter-----		Release	Number of		
	Local	LAN	WAN	CPU Util	Trans/Hr	Level	Processors		
1Quarter	.6	2.2	.0	4.3	2482	V4R4M0	4/4		
2Quarter	.6	2.2	.0	5.2	2979	V4R4M0	4/4		
3Quarter	.6	2.2	.0	6.2	3575	V4R4M0	4/4		
4Quarter	.6	2.2	.0	7.5	4290	V4R4M0	4/4		
5Quarter	.6	2.2	.0	9.0	5147	V4R4M0	4/4		
6Quarter	.6	2.2	.0	10.8	6177	V4R4M0	4/4		
7Quarter	.6	2.2	.0	12.9	7412	V4R4M0	4/4		
More...									
F3=Exit F10=Re-analyze F11=Alternative view F12=Cancel									
F15=Configuration menu F17=Analyze multiple points F24=More keys									
BEST/1 CPU model 730 2C6B refers to IBM CPU model 730 2067-1511									

Figure 369. Display Analysis Summary (Part 1 of 2)

Display Analysis Summary									
Period	CPU Model	Stor (MB)	CPU Util	-Disk Nbr	IOPs-- Util	-Disk Ctls-- Nbr	Util	-Disk Arms-- Nbr	Util
7Quarter	730 2C6B	16384	12.9	3	6.2	28	.2	40	2.4
8Quarter	730 2C6B	16384	15.5	3	7.5	28	.2	40	2.9
9Quarter	730 2C6B	16384	18.6	3	9.0	28	.3	40	3.5
10Quart	730 2C6B	16384	22.4	3	10.7	28	.3	40	4.2
Bottom									
Period	--Non-Inter Rsp Time--			-----Non-Inter-----		Release	Number of		
	Local	LAN	WAN	CPU Util	Trans/Hr	Level	Processors		
8Quarter	.6	2.2	.0	15.5	8895	V4R4M0	4/4		
9Quarter	.6	2.2	.0	18.6	10674	V4R4M0	4/4		
10Quart	.6	2.2	.0	22.4	12809	V4R4M0	4/4		
Bottom									
F3=Exit F10=Re-analyze F11=Alternative view F12=Cancel									
F15=Configuration menu F17=Analyze multiple points F24=More keys									
BEST/1 CPU model 730 2C6B refers to IBM CPU model 730 2067-1511									

Figure 370. Display Analysis Summary (Part 2 of 2)

21.5.18 High priority threshold

BEST/1 defaults to making its CPU utilization recommendations and conclusions based on CPU utilization of priority 20 or higher jobs. Many server jobs, such as QZDAINIT, run at priority 20 by default.

If your server jobs run at lower priorities (value greater than 20) and you want these to be included in values used by BEST/1, you can modify the guidelines and thresholds as follows:

1. Select option **60** for More BEST/1 Options.
2. Select option **11** for the Analysis parameter menu.
3. Select option **1** for Edit utilization guidelines.

The display shown in Figure 371 appears.

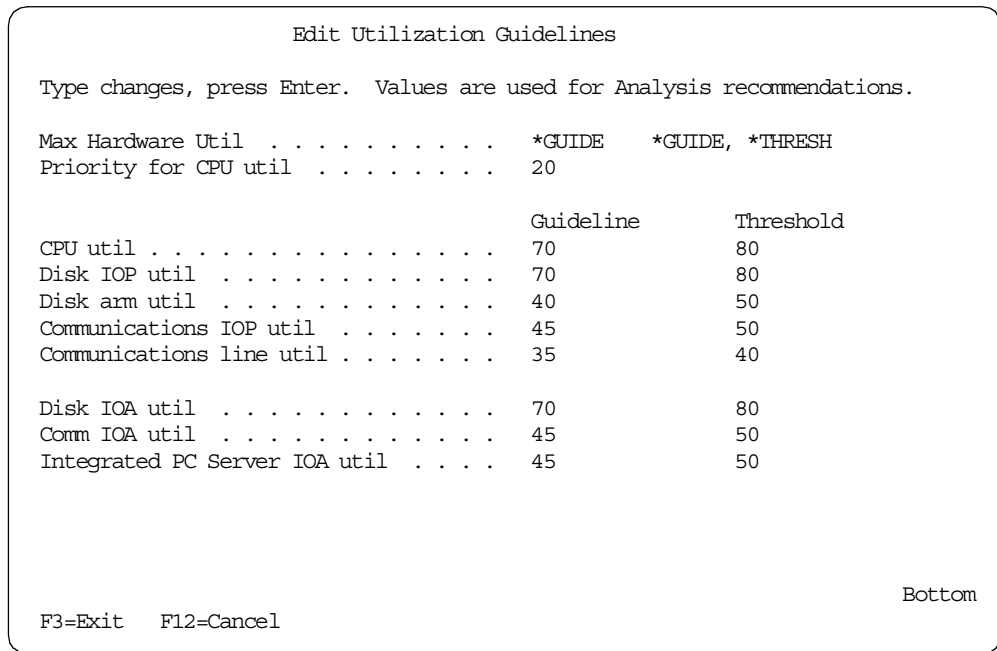


Figure 371. Edit Utilization Guidelines

You can change the required guideline and threshold values on this display.

Table 52 shows the guidelines for CPU utilization for various n-way processors of the AS/400 system.

Table 52. Guidelines for CPU utilization by number of processors

Number of processors	CPU utilization
1-way	70%
2-way	76%
4-way	81%
8-way	89%
12-way	92%

21.5.19 Other considerations

The capacity planning methodology presented here focuses on gathering a relevant snapshot of real data obtained from our Web application environment for the client, server, and network environments. It also looks at correlating it with user-based requests, such as Web hits and transactions. This obviously

represents the past, but we can learn from it by predicting the future for our particular environment. We have also indirectly dealt with the complex topic of forecasting future growth, but with few specifics outside of plugging in uplift factors into the BEST/1 capacity planning tool.

21.5.20 Growth

The most important factors in capacity planning deal with predicting future workloads and application environments for the Web site. Whether you plan on 5% growth per year or per hour has a major impact on the server and network infrastructure. It also affects the funds and other resources that you request from your organization's financial group. The resources that you request and what you receive are often quite different.

Our intent is not to recommend complex mathematical forecasting techniques, such as exponentially weighted averages or modeling seasonal effects with sinusoidal or other goodness of fit tests, which often have minimal practical benefit. By the same token, blindly relying on a growth rate of x% because that's what a favorite computer magazine says is also usually not valid. The most beneficial forecasting correlates non-random organization events (earnings reports, new products or services available, marketing campaigns or product reviews in the trade press, holiday shopping season) to changes on the server. Let's look at an example of transactions over a fairly lengthy period of time (Figure 372).

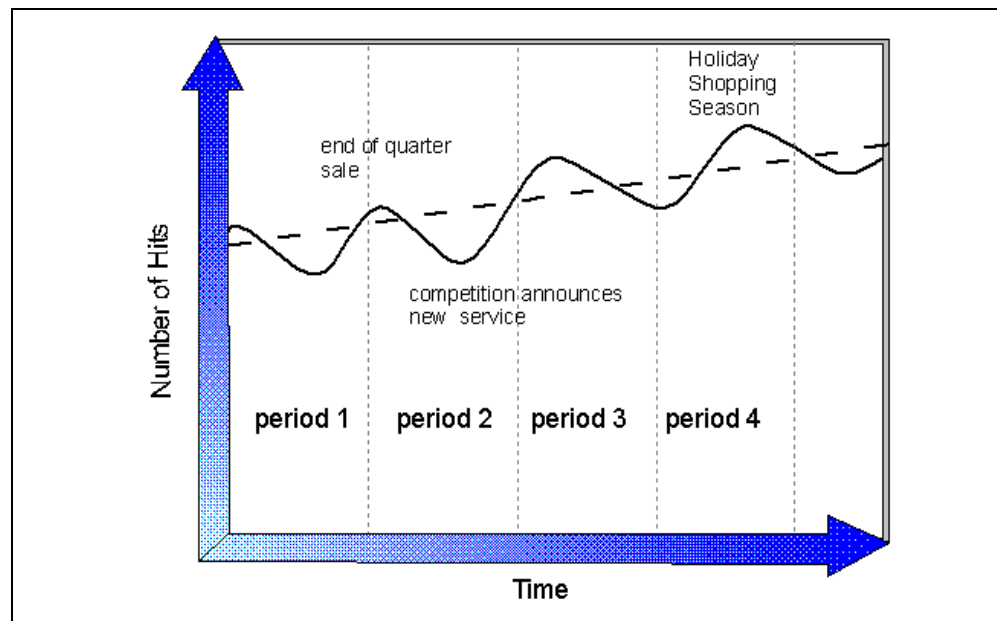


Figure 372. Measured Web site hits per day over time

You will generally see a great variation in transactions per day. Tuesdays may be your most busy day, peak traffic tends to occur at 4:00 p.m. Eastern time, and so on. Often times peak and trough conditions occur because of predictable events, or a lack thereof. Also, we have drawn a linear goodness of fit through the data for a rough idea of the growth rate.

Besides short term and long term growth, you also need to understand the peak load for which you need to plan. As we have seen numerous times, looking at

averages is easy, but there may be explicable and inexplicable situations where an overall average of 10 transactions per second and 10% CPU utilization periodically reaches 20 transactions per second and 30% CPU utilization.

21.5.21 Summary

Capacity planning involves careful analysis of the server and network resource load. You must be able to use this data and application transaction data to determine the necessary resources to meet a given service level. For your particular Web application environment, you need a thorough understanding of the actual transactions.

We also showed a number of examples using the AS/400 Performance Monitor data to understand key server resource usage (CPU, disk arms, main memory, and communications IOPs). The BEST/1 tool can be used to model any number of future scenarios for OneWorld application endeavors.

Capacity planning for network resources is a complex subject in and of itself. You may need to consult with network specialists, especially if you choose to redesign or change the architecture of your communications infrastructure.

Each of these topics, individually, has a bearing on the overall key metric, response time. You may want to pursue an application monitoring solution or employ a third-party monitoring service to accurately and objectively track this metric. Also, look at the multiple-tier Application Server architecture closely.

Keep in mind that OneWorld application data collection and analysis and capacity planning techniques *are* an ongoing business process (Figure 373). They are not a once-in-a-while event that is done grudgingly. Your application environment will be subjected to constant changes, uncertainty, and factors outside of your control. You do not want to log every request to your server and may not have the AS/400 Performance Monitor running at all times, as your system matures. Rather, periodic sampling of these and other appropriate statistics is a better choice for striking the fine balance between too much and too little data.

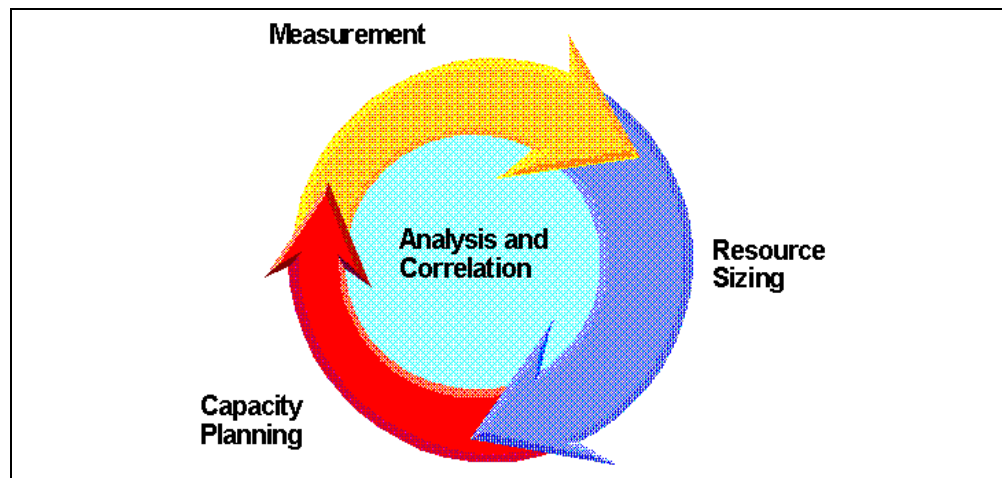


Figure 373. Measurement, analysis, correlation, sizing, and capacity planning

In other words, capacity planning is a journey rather than a destination. It also is one of several essential ingredients for having a successful IT environment that performs at a price your management can accept.

Chapter 22. Problem identification and solution

This chapter is intended to help you identify and manage problems that you may encounter in running OneWorld on the AS/400 system. It focuses on identifying the source of problems and suggests possible solutions. The chapter does not cover all situations or problems that you could encounter. However, it should give you an idea of where and how to start identifying a solution for your problem.

The J.D. Edwards *OneWorld Server and Workstation Administration* (B73.3.2) manual contains detailed information on troubleshooting the OneWorld client and Enterprise Server.

22.1 Problem determination checklists

The following common categories are where most customers encounter problems while running J.D. Edwards OneWorld on the AS/400 system:

- Performance
- Upgrades
- Installation
- OneWorld and WorldSoftware coexistence

For each of these areas, there is a section in this chapter that provides a set of questions to help further identify the specific problem. Once the problem is identified, some possible solutions are suggested in the checklist section.

22.1.1 General problem determination checklist

Table 53 is a general checklist you can use to diagnose typical problems before calling software support.

Table 53. Problem determination checklist

X	Follow these steps to isolate a perceived problem:
	Enter the command <code>DSPMSG QSYSOPR</code> to display QSYSOPR message queue. If a message has an asterisk (*) next to it, you can run the Work with Problems command by placing the cursor on the line and pressing the F14 key.
	To go directly to the Work with Problems screen and display system-detected problems, enter the <code>WRKPRB</code> command. From this screen, you can display and work with potential problems.
	Use the <code>DSPPTF OUTPUT(*PRINT)</code> command to generate a list of all of the PTFs on your system to a spooled file (QSYSPRT). You need to verify that you have all of the appropriate PTFs on your system. This allows you to use the Find feature to search the file to locate specific PTFs in the following steps.
	Verify that your system is at the latest OS/400 Cumulative PTF level. Use the command <code>SNDPTFORD SF98VRM</code> (where V=Version R=Release and M=Modification, for example, SF98430) to pull down the latest CUM PSP information. After it is downloaded, you can use the <code>DSPPFM FILE(QGPL/QAPZCOVER) MBR(QSF98vVRM)</code> command to view the file and compare the latest CUM version to the CUM version on your system. To order the latest version through ECS, enter <code>SNDPTFORD SF99VRM</code> . Check your version and service pack level of OneWorld. From the OneWorld Explorer, select Help and About OneWorld .

X	Follow these steps to isolate a perceived problem:
	Verify that you have the Database Fix pack installed on your system and that it is at the latest release. Enter the DSPDTAARA QSYS/SF9910x command, where x is your release of OS/400, to view the Group PTF # (for example, SF99103-03 V4R3M0 01/27/1999). New DB Fix packs are usually released about every two months. If the date listed in the data area is not recent, use the SNDPTFORD SF9910x command to order the latest Fix pack. It will be mailed on your default media. Check the cover letter to compare with your release.
	Verify that you have the required J.D. Edwards OneWorld PTFs applied on your system. Some are listed in the J.D. Edwards OneWorld Installation Guide. To ensure that you have the most current version, check one the following Informational APARs: II11676 (for V4R2) or II11677 (V4R3). To order through ECS, use the command SNDPTFORD II11676.
	Gather information on QSERVER jobs for support by using the following commands: DSPACTPJ SBS (QSERVER) PGM (QIWS/QZDASOINIT) OUTPUT (*PRINT) DSPACTPJ SBS (QSYSWRK) PGM (QIWS/QZSOSIGN) OUTPUT (*PRINT) DSPACTPJ SBS (QSERVER) PGM (QIWS/QZSCSRVS) OUTPUT (*PRINT)
	If you are running J.D. Edwards WorldSoftware and OneWorld in a coexistent environment, ensure that you have the required WorldSoftware cumulative PTFs installed. For example, OneWorld Release B73.3 requires that WorldSoftware be at Release A7.3 with Cumulative PTF update 9 or greater, and coexistence PTF A73PC000X0 must be applied. To verify the WorldSoftware PTF level that is running, go to the Software Upgrade Menu (A97IBM) and select option 13 , Inquire on Upgrade History.
	Identify the other applications that are running on the system, and verify that they are set up properly referring to the appropriate documentation.
	Verify that CNC is set up properly and be prepared to explain your environment to software support (if necessary). Refer to the J.D. Edwards <i>OneWorld Configurable Network Computing Implementation Guide</i> for assistance.
	Verify the version and service level of the Client Access ODBC driver. To display it, click the Windows Start menu. Highlight Settings , and click Control Panel . Double-click the Client Access icon to display the General Properties. Ensure that you have the latest Service Pack applied. You can view (and download) the latest Service Packs at: http://www.as400.ibm.com/clientaccess/casp.htm You can also view the Client Access Informational APARs at: http://www.as400.ibm.com /clientaccess/caiixd1.htm

22.1.2 Problems that affect system and application performance

Performance problems may be experienced system wide or application wide. This section contains a list of most common questions that you should ask and a checklist of activities you should perform (Table 54 and Table 55 on page 714).

22.1.2.1 AS/400 system problems

Review the following scenarios. Considering the problem that is occurring with your system, refer to Table 54 for an explanation of the numbers that are listed. Each number corresponds to a particular action to help correct the problem:

- Is this problem effecting system-wide performance? If yes, see checklist items 1, 2, 3, 6, 7, 8, and 14.
- Is this problem effecting only a subsystem's performance? If yes, see items 1, 2, 3, 4, 5, 6, 7, 8, 10, and 11.

- Is this problem effecting an applications performance? If yes, see items 1, 3, 4, 5, 13, 15, 16, 19, 20, and 21.
- Is this problem effecting a transaction within the application? If yes, see items 1, 3, 5, 13, 17, 18, 20, and 21.
- Is this problem effecting a jobs performance? If yes, see items 1, 3, 5, 13, 16, 20, and 21.

Table 54. AS/400 performance problems checklist

Item #	Suggested activity: AS/400 command to use and activity to perform
1	Check the system status (WRKSYSSTS). See "Work with System Status (WRKSYSSTS)" on page 570.
2	Check the % of disk being used (WRKSYSST). See "Work with System Status (WRKSYSSTS)" on page 570.
3	Is there one job or subsystem taking a lot of CPU % (WRKACTJOB). See "Work with Active Job (WRKACTJOB)" on page 573.
4	Check the status of the subsystem (WRKACTJOB). See "Work with Active Job (WRKACTJOB)" on page 573.
5	Check the job log of the job in question (WRKJOB). See 22.3.1, "Job identification" on page 719.
6	Check the system value QPFRADJ (WRKSYSVAL). See "Work with System Values (WRKSYSVAL)" on page 569.
7	Check the % of CPU being used (WRKSYSSTS). See "Work with System Status (WRKSYSSTS)" on page 570.
8	Check the system value QMCHPOOL (WRKSYSVAL). See "Work with System Status (WRKSYSSTS)" on page 570.
9	Check the disk activity (WRKDSKSTS). See "Work with Disk Status (WRKDSKSTS)" on page 572.
10	Check the current settings of Prestart job QZDASOINT. Enter DSPSPBSD QSERVER Select option 10 and then option 10 and option 5 for QIWS/QZDASOINIT.
11	Check the current, average and peak number of prestart jobs: DSPACTPJ QSERVER QIWS/QZDASOINIT
12	Check the system value QQRYDEGREE (WRKSYSVAL). See "Work with System Values (WRKSYSVAL)" on page 569.
13	Check to see if this problem occurs during a specific time during the work day (DSPJOBLOG). See 22.3.1, "Job identification" on page 719.
14	Check the faulting levels on the system (WRKSYSSTS). See "Work with System Status (WRKSYSSTS)" on page 570.
15	Collect Performance Monitor data (STRPRFMON). See 21.3.3.1, "Collecting Performance Monitor data" on page 606.
16	Check to see what program is running (WRKACTJOB). See "Work with Active Job (WRKACTJOB)" on page 573.
17	Check to see what the transaction is doing (PRTTNSRPT). See 21.3.4.2, "Performance Tools/400 reports" on page 610.
18	Check the SQL package information (PRTSQLINF). See 21.3.5.3, "Print SQL Information" on page 621.

Item #	Suggested activity: AS/400 command to use and activity to perform
19	Use the tool DBMON (STRDBMON). See "Collecting communications trace" on page 661.
20	Check to see if the application/transaction is running on the client or server. If running on the client, debug the problem on the client. See 22.5.8, "Client Access Express diagnostics" on page 767.
21	Check to see if the environment is running on a LAN or WAN. A WAN environment is slower than a LAN environment. Perform the action in item 22 to see if the slow down is in the network.
22	Run a communications trace against the communications line being used by the application or transaction during the problem occurrence (STRSST or STRCMNTRC). See 22.3.4.1, "AS/400 communications trace" on page 746.

22.1.2.2 OneWorld problems

Review the following scenarios. Considering the problem that is occurring with your system, refer to Table 55 for the numbers that are listed. Each number corresponds to a particular action to help correct the problem.

- Is this an issue with submitting UBEs? If yes, see checklist items 1, 2, 3, 4, 5, 6, and 8.
- Is this an issue with processing UBEs? If yes, see items 1, 2, 3, 4, 5, and 6.
- Is this an issue with UBEs completing successfully? If yes, see items 1, 2, 3, 4, 5, 6, and 7.

Table 55. OneWorld performance activity checklist

Item #	Suggested OneWorld activity
1	Make sure there is enough available DASD.
2	Go to the J.D. Edwards Knowledge Garden and verify that the minimum technical requirements have been met (http://www.jdedwards.com).
3	Go to the J.D. Edwards Knowledge Garden and check "Breaking News" (http://www.jdedwards.com).
4	Go to the IBM J.D. Edwards support Web sight to ensure the proper PTFs have been applied (http://www.as400.ibm.com/service/bms/jde-support.htm).
5	Turn on debugging in the jde.ini (client) or the INI (Enterprise Server). Make sure that the cause of the issue is not clearly explained in the jde.log or jddebug.log. (save these logs in case it becomes necessary to contact response line).
6	Make sure that the JDEB7332 subsystem starts up successfully (specifically the NETWORK and SENTINEL jobs stay up in a "wait" status).
7	Verify the "System – B733" ODBC Data Source is working properly.
8	Verify the *Public object authority for the user profile that is responsible for submitting the UBE on the AS/400 system is set appropriately.

22.1.3 Problems with upgrades

When upgrading OS/400 releases, the system values are saved across the upgrade. However, some settings are not saved and therefore should be documented prior to the upgrade and then set again after the upgrade. Perform

the activities in Table 56. Where item 1 should be done before the upgrade, other items are done after the upgrade.

Table 56. AS/400 upgrade activity checklist

Item #	Suggested activity: AS/400 command to use and details on activity
1	Check the current settings of Prestart job QZDASOINT. Enter: DSPSBSD QSERVER Select option 10 and then option 10 and option 5 for QIWS/QZDASOINIT.
2	Change the settings for the QZDASOINIT prestart jobs in the QSERVER subsystem back to what they were before the upgrade (CHGPJE).
3	Delete User SQL packages to ensure that enhancements in the database optimizer are reflected in the SQL package.
4	Measure and monitor system performance and make possible tuning changes that may be needed because of enhancements in the new release.
5	PTFs for the tool DSKBAL are available in V4R1 thru V4R3 to assist in balancing data across disk drives. In V4R4, the STRASPBAL command is implemented to provide more functionality. This is important when new disk drives are added. See the V4R4 reference manuals or the help text for the command for options on when and how to balance.

The OneWorld upgrade process is described in the *OneWorld Upgrade Guide (AS/400 Systems)*. Prior to attempting the OneWorld upgrade, perform the activities in Table 57.

Table 57. OneWorld upgrade activity checklist

Item #	Suggested OneWorld activity
1	Make sure all OneWorld customized and modified objects have been checked in.
2	Make sure the Deployment Server and workstations are at the proper MDAC level (check the J.D. Edwards Web sight at: http://www.jdedwards.com).
3	Make sure that all of the appropriate AS/400 PTFs have been applied (http://www.as400.ibm.com/service/bms/jde-support.htm)
4	Make sure there is enough disk space to handle the new objects (disk space requirements are outlined in the <i>OneWorld Upgrade Guide (AS/400 Systems)</i>).

22.1.4 Installation problems

Review the following scenarios, and refer to Table 58 on page 716 for the listed numbers. Each number corresponds to a particular action to help correct the problem.

- Is this an issue with installing OneWorld on the Deployment Server? If yes, see items 1, 2, 3, 4, and 10.
- Is this an issue with Release for the Installation Plan? If yes, see items 1, 2, 3, 4, 5, 7, and 10.
- Is this an issue with Installation Workbench? If yes, see items 1, 3, 4, 5, 7, 8, 10, and 11.
- Is this an issue with PORTTEST? If yes, see items 3, 4, and 5.
- Is this an issue with the OneWorld services starting in the JDEB7332 Subsystem? If yes, see items 3, 4, 5, and 13.

- Is this an issue with Signon Security? If yes, see items 3, 4, 5, 6, 13, 14, and 15.
- Is this an issue with installing OneWorld on a client? If yes, see items 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, and 15.
- Is this an issue with logging into OneWorld on a client? If yes, see items 3, 4, 5, 6, 7, 8, 10, 13, and 14.

Table 58. OneWorld installation activity checklist

Item #	Suggested OneWorld activity
1	Make sure the Windows User ID (specifically JDE on the Deployment Server) is a member of the "Administrators" group in Windows security.
2	Make sure there is enough available disk space.
3	Go to the J.D. Edwards Knowledge Garden and verify that the minimum technical requirements have been met (http://www.jdedwards.com).
4	Go to the J.D. Edwards Knowledge Garden and check "Breaking News" (http://www.jdedwards.com).
5	Go to the IBM J.D. Edwards support Web sight to ensure the proper PTFs have been applied (http://www.as400.ibm.com/service/bms/jde-support.htm).
6	Make sure that the JDEB7332 subsystem starts up successfully (specifically the NETWORK and SENTINEL jobs stay up in a "wait" status).
7	Make sure there is a defined Client Access connection to the Enterprise Server.
8	Verify the System – B733, ODBC Data Source works properly.
9	Make sure the B7332 directory on the Deployment Server is properly shared.
10	Ensure the appropriate MDAC level has been applied to the Deployment Server and all workstations running OneWorld (check the J.D. Edwards Web sight http://www.jdedwards.com).
11	Make sure the Client Access level is the same as the Deployment Server.
12	Make sure the JDE user profile has the appropriate special authority.
13	Turn on debugging in the INI (Enterprise Server). Make sure that the cause of the issue is not clearly explained in the jde_#####.log or jdedebug_#####.log (save these logs in case it becomes necessary to contact response line).
14	Turn on debugging in the jde.ini (workstation). Make sure the cause of the issue is not clearly explained in the jde.log or jdedebug.log (save these logs in case it becomes necessary to contact response line).
15	Make sure the OneWorld System User ID and Password has been entered correctly in OneWorld User Security (P98OWSEC). If no OneWorld System User ID has been defined for the OneWorld User ID, OneWorld will try to make the connection with the OneWorld User ID. This means that the OneWorld User ID needs an AS/400 user profile with the same name as the OneWorld User ID and with appropriate authority.

22.1.5 Problems related to OneWorld and WorldSoftware coexistence

Review the following scenarios. Considering the problem that is occurring with your system, refer to Table 59 for the numbers that are listed. Each number corresponds to a particular action to help correct the problem.

- Did you install WorldSoftware before installing OneWorld Software? If *no*, see checklist item 1.
- Was WorldSoftware installed prior to OneWorld? If yes, see item 1.
- Are you getting “Level Check” errors? If yes, see item 2.

Table 59. OneWorld co-existence activity checklist

Item #	Suggested OneWorld activity
1	Install J.D. Edwards WorldSoftware (Release A73 CUM 9 and A73PC000X2 Coexistence update).
2	The table being accessed is in a OneWorld format and needs to be in a WorldSoftware format. This happens when a WorldSoftware table was dropped and then re-created from OneWorld.

22.2 Problem prevention

This section describes where to find information on how to prevent problems.

22.2.1 Informational APARs

IBM maintains various informational APARs (InfoAPARs) that detail specific PTFs required or recommended to run OneWorld and other useful information. They can be found on the Internet at:

<http://as400service.rochester.ibm.com/supporthome.nsf/Document/10000035>

Once you reach the site, select **All InfoAPARs by Release**. Then, select the Search function, and type `oneworld`.

Currently, OneWorld Informational APARs for specific releases of OS/400 include:

- II11751 - V4R1M0
- II11676 - V4R2M0
- II11677 - V4R3M0
- II11881, II12247 - V4R4M0

In addition, a link is provided at the following site so that the customer can subscribe to be notified by e-mail when the Informational APARs for OneWorld have been updated: <http://www.as400.ibm.com/service/bms/jde-support.htm>

The following Informational APARs pertain to Terminal Server Support with Client Access Express and AS/400 NetServer:

- II11373: Microsoft Windows NT Server 4.0, Terminal Server Edition
- II11435: AS/400 NetServer and Microsoft Windows NT Server 4.0, Terminal Server
- II11655: Client Access Connection Problem - Message CWBCO1098
- II11851: Client Access Support for Multi-processor PCs and Multi-threading

The Informational APAR that pertains to Integrated Netfinity Server is II10739 Version 4 Release 2 Windows NT on the IPCS Additional Information and PTFs.

22.2.2 Preventive Service Planning (PSP)

IBM maintains various Informational APARS and documents that detail specific PTFs that are required or recommended to run on the AS/400 system, as well as other useful information. They can be found at:

<http://as400service.rochester.ibm.com/supporthome.nsf/Document/10000031>

This Web site is labeled AS/400 Preventive Service Planning Information. On this page, you can find the following categories and information in these categories:

- All Preventive Service Planning Documents
 - Current Cumulative PTF Package
 - Summary of Hiper and Defective PTFs
 - HIPERS
 - Defective PTFs
 - Software Installation Information
 - Hardware Information
 - Previously Supported CUM PTF Package
 - Cumulative PTF Package Instructions
 - System Support Program (SSP) (Advanced 36)
 - Summary Listings
 - Read This First and Memo to Users
 - Upgrade Road Map Corrections (by page sequence)
 - Group PTFs
 - Cumulative PTF Package Coverletter Listings
 - Userlet (User Letter Publications)
- All Preventive Service Planning Documents by Release
 - R440
 - R430
 - R420
 - R410
 - R370
 - R360
 - R320
- All User (Memo to Users) Letter by Release
 - R420
 - R410
 - R370
 - R320
 - R075
- All Group PTFs by Release
 - R440
 - R430
 - R420
 - R410
 - R370
 - R360
 - R320
- Y2K PTF information: Information about Y2K PTFs
- Search PSP Information: Allows you to search all the categories by topic

22.3 Problem identification aids

This section describes aids that can help you identify problems, for example, commands that help you identify your jobs on the system and how to use logs, messages, and traces.

22.3.1 Job identification

To identify a problem of a certain job, you need to identify that job in the system first.

Every job on the AS/400 system is uniquely identified by a job ID (identification), which is comprised of job name, user ID, and job number. A job number is what is really unique. The following AS/400 commands can be used to identify the AS/400 jobs on which you want to work:

- **WRKJOB:** The Work with Job command produces a display like the example shown in Figure 374.

```
                                Work with Job
                                System:  ASM23
Job:  QZDASOINIT   User:  QUSER           Number:  018568

Select one of the following:

    1. Display job status attributes
    2. Display job definition attributes
    3. Display job run attributes, if active
    4. Work with spooled files

    10. Display job log, if active or on job queue
    11. Display call stack, if active
    12. Work with locks, if active
    13. Display library list, if active
    14. Display open files, if active
    15. Display file overrides, if active
    16. Display commitment control status, if active

                                                                More...

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
```

Figure 374. Sample display of the WRKJOB command

You see this display when you select option 5 (Work with) to work with a job on any other job-related commands such as WRKUSRJOB, WRKOBJLCK, WRKSBMJOB, WRKACTJOB, and so on.

The Work with Job command allows you to select options to work with or to change information related to a user job. You can group information that you can display into two types, depending on the status of the job and whether it is active. You can see information about the following options regardless of where the job is located in the system (on a job queue, on an output queue, or active):

- Job status attributes
- Job definition attributes
- Spooled file information

Information about the following options can be seen only when the job is active:

- Job run attributes
- Job log information
- Call stack information
- Job lock information
- Library list information
- Open file information
- File override information
- Commitment control status
- Communications status
- Activation groups
- Mutexes
- Threads

Note: A job is not considered to be in the system until all of its input is completely read in. Only then, is an entry placed on the job queue.

- **WRKUSRJOB:** The Work with User Jobs display (Figure 375) shows the names and status information of user jobs running in the system. It also shows user jobs that are on job queues or output queues. You can type an option next to one or more jobs. When the Enter key is pressed, the function associated with the number you typed is performed for that job.

```
Work with User Jobs                                08/03/98 11:31:07
Type options, press Enter.
 2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
 8=Work with spooled files 13=Disconnect

Opt  Job          User      Type    -----Status-----  Function
   QPADEV0005  GUEST    INTER   ACTIVE                  CMD-WRKUSRJOB
   QPADEV0007  GUEST    INTER   OUTQ
   QPADEV0007  GUEST    INTER   OUTQ
   QPADEV0008  GUEST    INTER   OUTQ
   QPADEV0009  GUEST    INTER   OUTQ

Parameters or command                                Bottom
====>
F3=Exit      F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display schedule data
F12=Cancel   F21=Select assistance level
```

Figure 375. Sample display for WRKUSRJOB command

As you can see in the Figure 375, WRKUSRJOB shows all the jobs of a selected user regardless of the job status.

- **WRKOBJLCK:** The Work with Object Locks (WRKOBJLCK) command allows you to work with the object lock requests in the system for a specified object. You can work with both held locks and locks waiting to be applied. You can type an option next to one or more jobs. When the Enter key is pressed, the function associated with the number you typed is performed for that job.

- **WRKSBJOB:** The Work with Submitted Jobs display shows the job names and status information of jobs submitted at a workstation, or submitted by a user. Jobs submitted with DSPSBJOB (*NO) specified on the Submit Job (SBJOB), Submit Database Job (SBMDBJOB), or Submit Diskette Job (SBMDKTJOB) commands are not shown on this display. You can type an option next to one or more jobs. When the Enter key is pressed, the function associated with the number you typed is performed for that job.
- **WRKSBSJOB:** The Work with Subsystem Jobs (WRKSBSJOB) display shows the names and status information of jobs processed by subsystems in the system, and of user jobs that are on job queues or output queues. You can type an option next to one or more jobs. When the Enter key is pressed, the function associated with the number you typed is performed for that job.
- **WRKACTJOB:** The Work with Active Jobs (WRKACTJOB) command allows you to work with performance and status information for the active jobs in the system.

22.3.1.1 Identifying client/server jobs

In OS/400 V4R4, you can use Operations Navigator to identify your server jobs by performing following steps:

1. Double-click the **Operations Navigator** icon.
2. If necessary, expand **My AS/400 Connections**.
3. Expand your AS/400 server.
4. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
5. Expand **Network**.
6. Expand **Servers**.
7. Click **TCP/IP** or **Client Access**, based on the type of servers for which you want to see jobs.
8. Right-click the host server for which you want to see jobs, and select **Server Jobs**.
9. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
10. Another window opens, showing the server jobs with the user, job type, job status, time entered system, and date entered system for that server.

The job name that is used on the AS/400 system consists of three parts:

- The simple job name
- The user ID
- The job number (ascending order)

The server job names follow these conventions:

- Job Name
 - For non-prestarted jobs, the server job name is the name of the server program.
 - Prestarted jobs use the name that is defined in the prestart job entry.
 - Jobs that are started by the servers use the job description name or a given name if they are batch jobs (the file server does this).

- The user ID
 - Is always QUSER, regardless of whether prestart jobs are used.
 - The job log shows which users have used the job.
- Work management creates the job number.

Two methods can be used to identify server jobs. The first method is to use the WRKACTJOB command. The second method is to display the history log to determine which job is being used by which client.

Identifying server jobs with the WRKACTJOB command

The WRKACTJOB command shows all active jobs, as well as the server daemons. You must press F14 to see the available prestart jobs.

These types of jobs are shown in Figure 376 and Figure 377:

- Server daemons, marked with (1)
- Prestarted server jobs, marked with (2)

```

                                Work with Active Jobs                                ASM23
                                03/11/99 08:39:42
CPU %:      .0      Elapsed time: 00:15:58      Active jobs: 152

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files  13=Disconnect ...

Opt Subsystem/Job  User      Type  CPU %  Function      Status
   QSYSWRK       QSYS     SBS    .0     DEQW          DEQW
   QZSCSRVS      QUSER    PJ     .0     TIMW          TIMW
   QZSCSRVS      QUSER    PJ     .0     TIMW          TIMW
   QZSCSRVSD     QUSER    BCH    .0     SELW          SELW
(1) QZSOSGND     QUSER    BCH    .0     SELW          SELW
(2) QZSOSIGN     QUSER    PJ     .0     PSRW          PSRW

Parameters or command
====>
F3=Exit  F5=Refresh  F7=Find  F10=Restart statistics
F11=Display elapsed data  F12=Cancel  F23=More options  F24=More keys

                                Bottom
  
```

Figure 376. Displaying client/server jobs with WRKACTJOB SBS (QSYSWRK)

```

Work with Active Jobs
ASM23
03/11/99 08:48:56
CPU %: .0 Elapsed time: 00:25:12 Active jobs: 152

Type options, press Enter.
2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
(2) QSERVER QSYS SBS .0 DEQW
(3) QPWFSEVSD QUSER BCH .0 SELW
QPWFSEVSO QUSER PJ .0 DEQW
QPWFSEVSO QUSER PJ .0 DEQW
QPWFSEVSO QUSER PJ .0 DEQW
QPWFSEVSO QUSER PJ .0 DEQW
QSERVER QPGMR ASJ .0 EVTW
QZDASVSD QUSER BCH .0 SELW
QZLSSERVER QPGMR BCH .0 EVTW

Parameters or command
Bottom

```

Figure 377. Displaying client/server jobs with WRKACTJOB SBS (QSERVER)

The following types of jobs are shown:

- ASJ (Autostart job for subsystem)
- SBS (Subsystem monitor job)
- BCH (Server daemon job)
- PJ (Prestarted server job)

Identifying server jobs using the history log

Each time a client user successfully connects to a server job, that job is swapped to run under the profile of that client user. To determine which job is associated with a particular user, you can display the history log with the Display Log (DSPLOG) command (Figure 378).

```

Display History Log Contents

.
.
Servicing job 017446/QUSER/QNPSEVSD for user JDE on 02/25/99 16:41:07 in subs
.
*SIGNON server job 017453/QUSER/QZSOSIGN processing request for user JDE on 0
Servicing job 017459/QUSER/QNPSEVSD for user JDE on 02/25/99 16:46:37 in subs
*SIGNON server job 017453/QUSER/QZSOSIGN processing request for user JDE on 0
Servicing job 017454/QUSER/QZDASOINIT for user JDE on 02/25/99 16:46:58 in su
*SIGNON server job 017453/QUSER/QZSOSIGN processing request for user JDE on 0
.
Servicing job 017455/QUSER/QZDASOINIT for user DLP on 02/25/99 16:46:59 in su
Job 017461/QUSER/QZDASOINIT started on 02/25/99 at 16:46:59 in subsystem QSER
Job 017462/QUSER/QZDASOINIT started on 02/25/99 at 16:46:59 in subsystem QSER
*SIGNON server job 017453/QUSER/QZSOSIGN processing request for user JDE on 0
Servicing job 017462/QUSER/QZDASOINIT for user JDE on 02/25/99 16:47:28 in su

```

Figure 378. Displaying the history log contents

Identifying server jobs for a user

To display the server jobs for a particular user, complete the following steps:

1. Double-click the **Operations Navigator** icon.
2. If necessary, expand **My AS/400 Connections**.
3. Expand your AS/400 server.
4. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.
5. Expand **Users and Groups**.
6. Click **All Users**.
7. Right-click on the user for whom you want to see server jobs.
8. Select **User Objects**.
9. Click **Jobs**.
10. If necessary, sign on with QSECOFR for User ID and the appropriate password for Password.

You see a window displaying all the server jobs for that user.

You can also use the WRKOBJLCK command to identify and work with your database server jobs (in this case, you need to know the user ID, such as QUSER, that was used to start Client Access) by performing the following steps:

1. Enter the following command:

```
WRKOBJLCK QUSER *USRPF
```

2. You see a list of jobs holding locks. Select the **QZDASOINIT** job.

There may be several QZDASOINIT jobs. You must try each one until you find the currently active job.

3. Enter option 5 for the QZDASOINIT job.
4. Enter option 10 from the Work with Job display to view the log.

22.3.1.2 OneWorld related jobs on the AS/400 system

There are three types of OneWorld jobs:

- ODBC database connection
- Kernel jobs
- Batch (UBE) jobs

OneWorld ODBC connections are serviced by *prestart* jobs in the QSERVER subsystem on the AS/400 system (refer to). Batch UBEs and OneWorld Kernel jobs run in the QBATCH subsystem or in any user defined batch subsystem.

A *prestart job* (PJ) is the AS/400 system's way to prepare a job to serve the client's request in advance. It is also a way to make an already initiated job reusable instead of going through the entire job initiation process, which can be quite extensive on the AS/400 system. When the system receives the service request from the client, it selects one of the prestarted jobs. In a OneWorld environment, make sure it has one. IBM ships a reuse value of 200, but you may want to monitor your prestarted jobs by using the following command:

```
DSPACTPJ SBS(QSERVER) PGM(QIWS/QZDASOINIT)
```


You will see a screen like the one shown in Figure 379.

```

                                Display Active Prestart Jobs
                                ASM23
                                03/11/99 14:58:03
Subsystem . . . . . : QSERVER      Reset date . . . . . : 03/04/99
Program . . . . . : QZDASOINIT   Reset time . . . . . : 09:00:34
Library . . . . . : QIWS         Elapsed time . . . . . : 0173:57:29

Prestart jobs:
Current number . . . . . : 7
Average number . . . . . : 1.4
Peak number . . . . . : 15

Prestart jobs in use:
Current number . . . . . : 5
Average number . . . . . : .4
Peak number . . . . . : 13

More...

Press Enter to continue.

F3=Exit F5=Refresh F12=Cancel F13=Reset statistics

```

Figure 379. Display Active Prestart Jobs

To find the maximum number of uses for a prestart job, enter the command:

DSPSPSD SBS (QSERVER)

Enter option 10 (Prestart Jobs), and select **QZDASOINIT** (Figure 380).

```

                                Display Prestart Job Entry Detail
                                System:  ASM23
Subsystem description:  QSERVER      Status:  ACTIVE

Program . . . . . : QZDASOINIT
Library . . . . . : QIWS
User profile . . . . . : QUSER
Job . . . . . : QZDASOINIT
Job description . . . . . : *USRPRF
Library . . . . . :
Start jobs . . . . . : *NO
Initial number of jobs . . . . . : 1
Threshold . . . . . : 1
Additional number of jobs . . . . . : 3
Maximum number of jobs . . . . . : *NOMAX
Maximum number of uses . . . . . : 200
Wait for job . . . . . : *YES
Pool identifier . . . . . : 1

More...

Press Enter to continue.

F3=Exit F12=Cancel F14=Display previous entry

```

Figure 380. Maximum number of uses for prestart job QZDASOINIT

Identifying OneWorld jobs, whether it is a prestart job or a batch job, is important in many tasks especially for problem determination and performance tuning. The next section explains how to identify OneWorld jobs on the AS/400 system.

OneWorld ODBC database connection jobs

On the AS/400 side, a OneWorld ODBC database connection job is realized as an ODBC server job, or rather a prestarted QZDASOINIT job in the QSERVER subsystem to be exact. A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from other jobs because they use prestart job entries to determine which program, class, and storage pool to use when they are started. Within a prestart job entry, you must specify attributes that the subsystem uses to create and manage a pool of prestart jobs.

Using prestart jobs allows you to reduce the amount of time required to handle a program start request. The initiation of a prestart job follows this procedure:

1. When a subsystem is started, or when the Start Prestart Job (STRPJ) command is entered, prestart jobs are started, based on the information contained in the prestart job entries.
2. When a program start request is received on the target system, it goes to the subsystem that has the required communications device allocated.
3. The program start request attaches to a prestart job that is already running if the subsystem finds either of the following entries:
 - A prestart job entry with a program name that matches the program name of a program start request
 - A routing entry that matches the routing data of the program start request and the routing program on the found routing entry that matches the program name on a prestart job entry

On a PC, the ODBC Database connections are created when the OneWorld workstation installation occurs. These ODBC Database connections are defined during the OneWorld Installation Workbench. This workbench takes the ODBC drivers on the Deployment Server to establish the ODBC settings that will be written to a PC. If the Deployment Server has a different MDAC level or Client Access level, there may be problems with establishing any or all ODBC connections from the PC to the OneWorld servers. ODBC Database connections are addressed in more detail in the Chapter 9, "OneWorld installation and maintenance" on page 151.

If the prestart job entry is not active, the program start request is rejected. If a match is not found, the program start request causes a communications batch job to start if the routing data matches the routing entry. Otherwise, the program start request is rejected.

OneWorld users typically use six to eight ODBC connections. Each ODBC connection is connected to one QZDASOINIT job. Finding the OneWorld job is tricky when an error occurs. There are several options available. Two are described here in more detail. You can choose the one you prefer most.

- **WRKSBSJOB QSERVER command**

Enter option 5 next to the job you "think" may be a OneWorld job (Figure 381). Unfortunately, there is no scientific way of telling which of these QZDASOINIT

jobs is a OneWorld job. If you have only one QZDASOINIT job and you are sure that you have at least one OneWorld job, that must be it.

```

                                Work with Subsystem Jobs
                                10/28/98 16:24:19
Subsystem . . . . . : QSERVER

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files  13=Disconnect

Opt Job          User          Type      -----Status-----  Function
   QSERVER      QPGMR      AUTO      ACTIVE
   QTFPJTCP     QUSER      PJ        ACTIVE
   QZDAINIT     QUSER      PJ        ACTIVE
  5  QZDASOINIT  QUSER      PJ        ACTIVE
   QZDASOINIT  QUSER      PJ        ACTIVE
   QZDASOINIT  QUSER      PJ        ACTIVE
   QZDASRVSD   QUSER      BATCH     ACTIVE

                                                                Bottom
Parameters or command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display schedule data
F12=Cancel

```

Figure 381. Work with System Jobs display

When you select option 5, the display shown in Figure 382 appears.

```

                                Work with Job

Job:  QZDASOINIT  User:  QUSER          Number:  015322

Select one of the following:

    1. Display job status attributes
    2. Display job definition attributes
    3. Display job run attributes, if active
    4. Work with spooled files

    10. Display job log, if active or on job queue
    11. Display call stack, if active
    12. Work with locks, if active
    13. Display library list, if active
    14. Display open files, if active
    15. Display file overrides, if active
    16. Display commitment control status, if active

                                                                More...

Selection or command
====> 1

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

```

Figure 382. Work with Job display

When you select option 1, a display appears like the example in Figure 383 on page 728.

```

                                Display Job Status Attributes
Job:  QZDASOINIT      User:  QUSER      Number:  015322

Status of job . . . . . :  ACTIVE
Current user profile . . . . . :  JDE
Entered system:
  Date . . . . . :  07/21/98
  Time . . . . . :  13:58:23
Started:
  Date . . . . . :  07/21/98
  Time . . . . . :  13:58:23
Subsystem . . . . . :  QSERVER
  Subsystem pool ID . . . . . :  1
Type of job . . . . . :  PJ
Special environment . . . . . :  *NONE
Program return code . . . . . :  0
Controlled end requested . . . . . :  NO
System . . . . . :  AS23
                                                                    Bottom

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F16=Job menu

```

Figure 383. Display Job Status Attributes display

Notice that Current User Profile is JDE. This indicates that this job is a OneWorld job. Also notice the job ID of this job, such as Job Name, User Name, and especially Job Number. You need this information for a number of occasions.

- **WRKOBJLCK OBJ(User ID) OBJTYPE(*USRPRF) command**

Here, *User ID* is the user profile used for the ODBC connection (Figure 384).

This option is documented in Informational APAR I109599 from the IBM Customer Support Center.

Tip

We suggest that you use an alternate user profile, for example, JDEDEBUG, when debugging issues that can be duplicated. This allows for fewer jobs to be checked when isolating the problem job.

```

Work with Object Locks
System:  ASM23
Object:  JDE      Library:  QSYS      Type:  *USRPRF

Type options, press Enter.
  4=End job  5=Work with job  8=Work with job locks

Opt   Job           User           Lock           Status
  5    QZDASOINIT    QUSER         *SHRRD        HELD
      QZDASOINIT    QUSER         *SHRRD        HELD
      QZDASOINIT    QUSER         *SHRRD        HELD
      QZDASOINIT    QUSER         *SHRRD        HELD
      QZDASOINIT    QUSER         *SHRRD        HELD
      QZSCSRVS     QUSER         *SHRRD        HELD
      QZSCSRVS     QUSER         *SHRRD        HELD
      QZSCSRVS     QUSER         *SHRRD        HELD
      QZSCSRVS     QUSER         *SHRRD        HELD
      QZSCSRVS     QUSER         *SHRRD        HELD

F3=Exit  F5=Refresh  F12=Cancel
Bottom

```

Figure 384. Results WRKOBJLCK OBJ(JDE) OBJTYPE(*USRPRF)

For User ID JDE, you see all the prestarted jobs and not every other user's job. If you have more than one user with the same user ID, this may become a little difficult. Enter option 5, Work with job. The display shown in Figure 385 appears.

```

Work with Job
System:  ASM23
Job:  QZDASOINIT  User:  QUSER      Number:  018886

Select one of the following:

  1. Display job status attributes
  2. Display job definition attributes
  3. Display job run attributes, if active
  4. Work with spooled files

 10. Display job log, if active or on job queue
 11. Display call stack, if active
 12. Work with locks, if active
 13. Display library list, if active
 14. Display open files, if active
 15. Display file overrides, if active
 16. Display commitment control status, if active

More...

Selection or command
===>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

```

Figure 385. Work with Job display

When you select option 1, the Display Job Status Attributes menu appears (Figure 386 on page 730).

```

                                Display Job Status Attributes
                                System:  ASM23
Job:  QZDASOINIT      User:  QUSER      Number:  018886

Status of job . . . . . :  ACTIVE
Current user profile . . . . . :  JDE
Job user identity . . . . . :  JDE
  Set by . . . . . :  *DEFAULT
Entered system:
  Date . . . . . :  03/10/99
  Time . . . . . :  08:55:18
Started:
  Date . . . . . :  03/10/99
  Time . . . . . :  08:55:18
Subsystem . . . . . :  QSERVER
  Subsystem pool ID . . . . . :  1
Type of job . . . . . :  PJ
Special environment . . . . . :  *NONE
Program return code . . . . . :  0

More...

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F16=Job menu

```

Figure 386. Job name, user ID, and job number

22.3.2 Logs

Every OneWorld job running in an AS/400 system subsystem has an associated job log. The job log for a job records those messages that were sent between the OneWorld job and the OS/400 operating system.

The job log is initialized when the job is started and remains in existence until the job ends. When a job ends normally, the job log may be deleted or written to a spooled file where it can be viewed or printed. When a job ends abnormally, a job log is usually produced. This is controlled by the message logging attributes specified for the job and the end code of the job. For example, if the message level value of the LOG parameter on the CHGJOB and CHGJOB commands is set to *NOLIST, a job log is not produced unless the end code is 20 or greater. A job end code of 0, which indicates a normal completion, does not usually produce a job log. To obtain a job log in any case, you can change the job descriptions used by the OneWorld jobs to a message level value for the LOG parameter to *SECLVL by using the command:

```
CHGJOB JOB(QSYS/QDFTJOB) LOG(4 00 *SECLVL) LOGCLPGM(*YES)
```

Note: QDFTJOB is the default job description. View the user profile to determine the job description name. Generally, QDFTJOB is used by IBM-supplied user profiles. We recommend that you change only non-IBM supplied job descriptions.

This change remains in effect after the next restart of OneWorld or the next time a job is started that uses this job description. During the installation of OneWorld, the installation jobs inherit the attributes from the job that started the installation. To make sure that a spooled file is generated, before starting or restarting the installation, change your interactive job using the command:

```
CHGJOB JOB(*) LOG(4 00 *SECLVL) LOGCLPGM(*YES)
```

While a job is running, use the WRKACTJOB, WRKOBJLCK, WRKSEMJOB, WRKUSRJOB, or any of the Work with Job type commands. Enter option 5, Work with, next to the desired job, and enter option 10, Display job log, if active or on job queue, to view the job log. This shows you high-level messages. To see all of the messages, press the F10 (Display detailed messages) key. If there is more than one page, press the Page Up key to see the beginning. Figure 387 shows an example of the information you may see.

```
Display Job Log
System: ASM23
Job . . . : QZDASOINIT   User . . . : QUSER       Number . . . : 018008

Job 018008/QUSER/QZDASOINIT started on 03/02/99 at 11:04:08 in subsystem
QSERVER in QSYS. Job entered system on 03/02/99 at 11:04:08.
ACGDTA for 018008/QUSER/QZDASOINIT not journaled; reason 1.
Servicing user profile JDE.

Bottom

Press Enter to continue.

F3=Exit   F5=Refresh   F10=Display detailed messages   F12=Cancel
F16=Job menu   F24=More keys
```

Figure 387. AS/400 Display Job Log

More detailed information is available for any of these messages by pressing the “help” function key (F1) while the cursor is over a message. The messages are explained in more detail in 22.3.3, “Messages” on page 738.

If you know the name of the job, you can also use the AS/400 system Display Job (DSPJOB) or Work with Job (WRKJOB) commands. The default action for the commands is to display the current interactive job. However, you can also enter the name of a OneWorld job such as JDENET_K. If only one job is found, you go to the Work with Job display. Here, you can select option 10 for the job log. Figure 388 on page 732 shows an example of the information you may see when you issue the WRKJOB command, and there is only one job active.

```

                                Work with Job
                                System: ASM23
Job:  JDENET_K      User:  ONEWORLD      Number:  017257

Select one of the following:

    1. Display job status attributes
    2. Display job definition attributes
    3. Display job run attributes, if active
    4. Work with spooled files

    10. Display job log, if active or on job queue
    11. Display call stack, if active
    12. Work with locks, if active
    13. Display library list, if active
    14. Display open files, if active
    15. Display file overrides, if active
    16. Display commitment control status, if active

More...

Selection or command
====> 10

```

Figure 388. Work with Job (WRKJOB) display for a single job

If multiple jobs are found, you see a display that lists all of them. The Status field tells you if the job is currently active or has ended, and there is only a job log on an output queue. Select option 1, Select, next to the job that you want. Figure 389 shows an example of the information you may see when you issue the WRKJOB command and there are multiple jobs.

```

                                Select Job
                                ASM23
                                03/02/99 13:05:48

Type option, press Enter.
  1=Select

Option Job      User      Number  Type  -----Status-----  Entered
  1  JDENET_K    ONEWORLD 017257 BCI    ACTIVE                 02/24/99
    JDENET_K    ONEWORLD 017255 BCI    ACTIVE                 02/24/99
    JDENET_K    ONEWORLD 017242 BCI    ACTIVE                 02/24/99
    JDENET_K    ONEWORLD 017241 BCI    ACTIVE                 02/24/99
    JDENET_K    ONEWORLD 017219 BCI    ACTIVE                 02/24/99

F3=Exit  F12=Cancel
Duplicate jobs found.

Bottom

```

Figure 389. Work with job (WRKJOB) display of multiple jobs

If a job is still active, enter the AS/400 system Display Job Log (DSPJOBLOG) command to view its job log. The default action for the command is to display the current interactive job. However, you can also give the qualified job name of a

OneWorld job. Every job on the AS/400 system has a job name, job user, and a unique six-digit number. In Figure 389, you can see the job name as jdenet_k, the job user as OneWorld, and the unique job numbers. If a job is still active and you want to create a spooled file or print out of the job log, use the following AS/400 system command:

```
DSPJOBLOG JOB (qualified job name) OUTPUT(*PRINT)
```

Use the AS/400 system Work with Spooled Files (`WRKSPLF`) command with `SELECT(*CURRENT)`, or the `WRKJOB` command with `JOB(*)` and option 4, Work with spooled files, to locate the generated spooled file.

If a job ends and you need to locate the spooled file for the job log, there are several ways to do this. One method is to use the `WRKSPLF` command and enter the user name that was associated with the job. In Figure 390, the user is OneWorld, so the `WRKSPLF SELECT(oneworld)` command can be used. Job logs can be identified by their name and by default. The name of the spooled file for a job log is `QPJOBLOG`. They are usually placed on the output queue `QEZJOBLOG`. If you know the name of the output queue, the AS/400 system Work with Output Queue (`WRKOUTQ`) command can also be used to find job logs.

```

                                     Display Spooled File
File . . . . . : QPJOBLOG                                     Page/Line 1/1
Control . . . . . :                                         Columns 1 - 130
Find . . . . . :
*.....1.....2.....3.....4.....5.....6.....7.....8.....9.....0.....1.....2.....3
5769SS1 V4R3M0 980729                                     Job Log          ASM23 03/05/99 12:42:19      Page 1
Job name . . . . . : QPADEV0009      User . . . . . : ONEWORLD      Number . . . . . : 018528
Job description . . . . . : ONEWORLD  Library . . . . . : QGPL      Section number . . . . . : 0001
MSGID  TYPE          SEV  DATE      TIME      FROM PGM  LIBRARY  INST  TO PGM  LIBRARY  INST
CPF1124 Information    00   03/05/99  12:42:05  QWTPLIPP  QSYS     05DF   *EXT  *N
Message . . . . . : Job 018528/ONEWORLD/QPADEV0009 started on 03/05/99 at
                  12:42:05 in subsystem QINTER in QSYS. Job entered system on 03/05/99 at
                  12:42:05.
*NONE   Command       03/05/99  12:42:05  QCADRV   QSYS     01A6   BV3C   QGPL    *SIMT
To module . . . . . : BV3C
To procedure . . . . . : BV3C
Statement . . . . . : 200
Message . . . . . : 200 - DSPOBJD OBJ(QSYS/B73*) OBJTYPE(*LIB)
                  OUTPUT(*OUTFILE) OUTFILE(QTEMP/QADSPBJD)
CPF9861 Information    00   03/05/99  12:42:05  QLIOUTFL QSYS     01C3   BV3C   QGPL    *SIMT
To module . . . . . : BV3C
To procedure . . . . . : BV3C
Statement . . . . . : 200
Message . . . . . : Output file QADSPBJD created in library QTEMP.
More...

```

Figure 390. `WRKSPLF` of Oneworld to get the job log

To view a job log spooled file, enter option 5 (display) next to the appropriate entry in the list. Figure 390 shows an example.

Another way to look at job logs is to use AS/400 Operations Navigator. In Figure 391 on page 734, all jobs are displayed. The `JDENET_K` job is selected by using right-clicking.

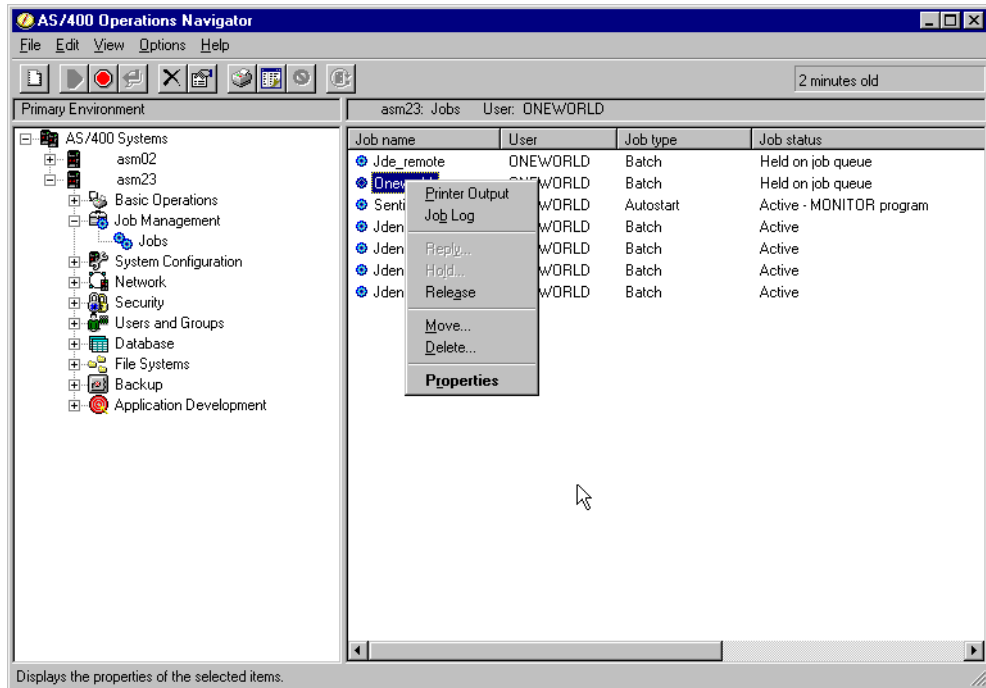


Figure 391. Operations Navigator job for user ONEWORLD

Note in Figure 391 that Properties is another option to choose when you right-click a job to display job attributes such as run priority and message logging.

Select the job log results and double-click the log that you want to view.

22.3.2.1 OneWorld logs

The primary tool for resolving OneWorld issues on any machine, including an AS/400 Enterprise Server, are JDE and JDEDEBUG logs generated by OneWorld. This section focuses on how to effectively use OneWorld logging on the AS/400 system.

AS/400 OneWorld log files are generated as stream files (STMF) in the integrated file system (IFS) based on the AS/400 INI file settings.

AS/400 Enterprise Server INI settings

There are some core INI settings that must be set appropriately to have OneWorld logs to be generated on the AS/400 system. These settings are:

```
[DEBUG]
Output=FILE (Send Messages to jdedebug.log)
Trace=FALSE
DebugFile=JDEB7332/jdedebug
JobFile=JDEB7332/jde.log
JDETSFile=/JDEB7332/JDETS.LOG
ClientLog=1 (Allow client log to record server log messages)
LogErrors=1 (Create jde.log and jdedebug.log)
KeepLogs=1 (Save UBE logs)
RunBatchDelay=0 (Delay the Batch job to edit code)
RepTrace=1 (Trace replication operations the jdedebug.log)
```

Here are some examples for the INI settings:

- **Example 1**

```
[DEBUG]
DebugFile=jdedebug
JobFile=jde.log
```

Log files generated in IFS root directory.

- **Example 2**

```
[DEBUG]
DebugFile=/jdeb733/jdedebug
JobFile==/jdeb733/jde.log
```

Log files generated in IFS directory /jdeb733

Note: The directory must exist with proper authority granted to the logging job.

To obtain more detailed information in the jde and jdedebug logs, the following INI settings can be modified:

- **[JDENET] netTrace=1**

netTrace focusses in on the OneWorld kernel jobs. Each kernel is responsible for specific processing. Chapter 7, “Work management” on page 109, discusses the kernel types. This setting allows for the tracing of the kernel processes. Possible values are:

- 0 Do not generate JDENet error messages (for example, communication between platforms).
- 1 Generate JDENet error messages.

- **[JDEIPC] ipcTrace=1**

IPC refers to Inter Process Communication. This a tool that OneWorld uses to allow each process the opportunity to know what other processes are doing. Using ipcTrace for OneWorld processes may be a useful troubleshooting tool. Possible values are:

- 0 Do not generate Inter-process Communication (IPC) error messages (for example, communication between processes on a single platform).
- 1 Generate IPC error messages.

- **[DEBUG] TAMTraceLevel=1**

TAM is set of database files in a binary-tree format which are copied from the data stored in Central Objects. These are typically referred to as runtime objects. Placing a trace on TAM could provide valueable information. Possible values are:

- 0 Do not generate Table Access Management (TAM) error messages (for example, regarding specification files).
- 1 Generate TAM error messages.

- **[UBE] UBEDebugLevel=6**

When UBE logging is turned on, OneWorld generates a UBE log with the same file name as the report it generates. The UBE logs are stored in the same place as the *.PDF files. The best tools to open these logs is either in Work with Servers (P986116) or Server Administration Workbench (P9861100). Possible values are:

- 0** Do not generate Universal Batch Engine (UBE) error messages.
- 1 through 6** Generate increasingly detailed error messages (for example, 1 gives the least specific messages, where 6 gives the most detailed messages).

- **[TCEngine] TraceLevel=10**

TraceLevel in the [TCEngine] section is specifically for logging issues with table conversion. Table conversions are most commonly part of the OneWorld Upgrade process. Since table conversions are usually run on the Deployment Server, the logs stay on the Deployment Server. These logs are held in the Printqueue directory. Possible values are:

- 0** Do not generate Table Conversion (TC) error messages.
- 1 through 10** Generate increasingly detailed error messages (for example, 1 gives the least specific messages, where 10 gives the most detailed messages).

Determining the log files location

OneWorld generates log files for the AS/400 system in the integrated file system. With IFS, OneWorld generates log files as stream files (STMF) in an IFS directory (DIR) based on the AS/400 INI file settings.

The INI file is located in the B733SYS library. You can find the log file locations in the [DEBUG] section of the file (Figure 392).

```

; OneWorld initialization file INI (JDE)
; AS/400 specific version - B73.3
; jde.ini file x:\oneworld\b733\hosts\as40\asm23

[DEBUG]
Output=FILE
Trace=FALSE
DebugFile=/JDEB733/jdedebug
JobFile=/JDEB733/jde.log
JDETSFile=/JDEB733/JDETS.LOG
ClientLog=0
LogErrors=1
KeepLogs=1

```

Figure 392. Log file settings in the INI file

Viewing log files from the AS/400 screen

There are two methods that can be used to view the OneWorld logs from the AS/400 system:

- Server Administration Workbench (SAW) on the AS/400 system, which is detailed in Chapter 7, “Work management” on page 109.
- Use the WRKLNK command to list the contents of the IFS directory tree and items contained within each directory.

Use the Display File (DSPF) command to view the IFS stream file, for example:

```
DSPF STMF('/JDEB733/jdedebug_132509.log')
```

You see a screen like the one shown in Figure 393.

```

Browse : jdeb733/jdedebug_017999.log
Record . : _____1 of 15 by __ 15           Column: __1 of 59 by __ 79
Control : _____
...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
Mar 2 10:50:32 ** **** jdeDebugInit -- output to file.
Mar 2 10:50:32 ** gxp_p_shm_hdl -2147483648 saw_gs_run_host <JDE>
Mar 2 10:50:34 ** API: jdeSawFreeEnv
Mar 2 10:50:34 ** Calling JDENET_FreeKernel
Mar 2 10:50:34 ** IPC2300009 - removeIPCmtx (idx 55) failed, error value =3021:
The value specified for the argument is not correct..
Mar 2 10:50:34 ** API: jdeSawInitEnv
Mar 2 10:50:34 ** Calling JDENET_InitKernel
Mar 2 10:50:34 ** IPC2100003 - Krnl4667RspQ already exists, and caller requeste
d exclusive access.
Mar 2 10:50:34 ** gxp_p_shm_hdl -2147483648 saw_gs_run_host <JDE>
Mar 2 10:50:34 ** API: jdeSawGetProcListV1( <JDE> , <6006> , ANY )
Mar 2 10:50:34 ** jdeSawGetProcListV1 : <JDE> <JDE>
Mar 2 10:50:34 ** API: jdeSawImpGetProcListV1( ANY )
Mar 2 10:50:34 ** ls_tmp <1> gi_max_kernel_proc 1

```

Figure 393. Viewing log files on the AS/400 system

You can view AS/400 OneWorld logs from a PC a couple of different ways. For more information, refer to the following section.

Viewing log files from the PC

There are several sets of OneWorld logs on the AS/400 system that can be viewed from a PC where OneWorld is installed:

- UBE logs can be viewed from within the OneWorld application, go to the Work with Server Jobs application (on the GH9011 menu). Select the **Server Map** data source. Highlight the appropriate job and choose **View Server Logs** from the Row menu.
- The Server Administration Workbench (P9861100) from the System Administration Tools menu (GH9011) can also be used to view OneWorld logs on an AS/400 server. See Chapter 7, “Work management” on page 109. There are even more details on SAW in the *OneWorld Server and Workstation Administration Guide*.
- A OneWorld log file in the AS/400 IFS can be accessed using Window Explorer.

We recommend that you use WORDPAD.EXE to view the log file. WordPad handles larger files better than other applications such as NotePad. Associate files with the “.log” extension with WordPad. Refer to Windows Help on how to set up file extension association.

Once the Client Access EBCDIC/ASCII conversion and file extension associations are set up, any AS/400 log file can be accessed by Windows Explorer as described here:

- a. From Windows Explorer, select **Tools->Map Network Drive**.
- b. Enter \\AS400Machine for the Path field, for example, \\ASM23 for accessing a log file on ASM23. Locate the mapped drive and the log file. Double-click the desired log file to view it (Figure 394 on page 738).

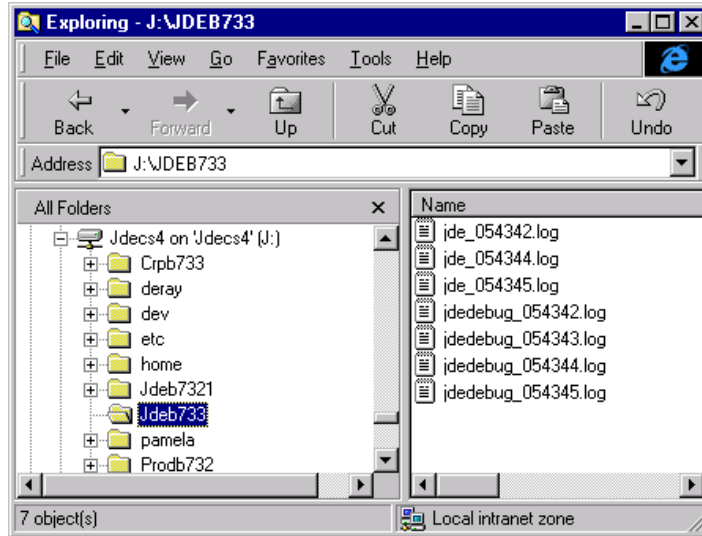


Figure 394. Accessing AS/400 IFS log files from Windows Explorer

22.3.3 Messages

On the AS/400 system, important information in the form of messages can be sent to message queues. To look at messages on a message queue, use the AS/400 system Display Messages (DSPMSG) command. There are both user message queues and system message queues.

If you select the default for the DSPMSG command Message Queue (MSGQ) parameter, you display the message queue for the user that you are currently signed on as. Figure 395 shows an example of the information you may see when issuing the DSPMSG command signed on as user ONEWORLD.

```

                                Display Messages
                                System: ASM23
Queue . . . . . : ONEWORLD           Program . . . . . : *DSPMSG
Library . . . . :   QUSRSYS           Library . . . . . :
Severity . . . . :   00               Delivery . . . . . : *NOTIFY

Type reply (if required), press Enter.
Job 017841/JDE/R014021 completed normally on 03/01/99 at 14:09:22.
Job 017842/JDE/R014021 completed normally on 03/01/99 at 14:11:19.
Job 017847/ONEWORLD/ONEWORLD completed normally on 03/01/99 at 14:30:41.
Job 017848/JDEINSTAL/JDEINSTAL completed normally on 03/01/99 at 14:31:53.
Job 017850/JDE/R04424 completed normally on 03/01/99 at 14:34:17.
Job 017855/JDE/R014021 completed normally on 03/01/99 at 14:43:17.
Job 017886/JDE/R014021 completed normally on 03/01/99 at 15:18:55.
Job 017894/JDE/R014021 completed normally on 03/01/99 at 15:29:40.
Job 017903/JDE/R0010P completed normally on 03/01/99 at 15:31:37.

```

Figure 395. DSPMSG ONEWORLD

If you change the message queue to QSYSOPR, you can view the messages sent from the AS/400 system to the system operator. Figure 396 shows an example of the information you may see when you issue the DSPMSG MSGQ (QSYSOPR) command.

```

                                Display Messages
                                System: ASM23
Queue . . . . . : QSYSOPR          Program . . . . . : *DSPMSG
Library . . . . : QSYS            Library . . . . . :
Severity . . . . : 90             Delivery . . . . . : *HOLD

Type reply (if required), press Enter.
reason code 0.
A mismatch exists between log file /home/web.logs/error.Q0990302 and log
format.
A mismatch exists between log file /home/web.logs/access.Q0990302 and log
format.
Performance monitor 017944/QPGMR/QPFRMON started at 00:01:28 03/02/99.
Device Q1PDEV cannot be varied off at this time.
Device Q1PDEV cannot be varied off at this time.
Adapter has inserted or left the ring on line TRNLINE.
Adapter has inserted or left the ring on line TRNLINE.
Adapter has inserted or left the ring on line TRNLINE.
* No cable was detected on communication port CMN01 of line
QESLINE.
Line QESLINE failed.
                                                                Bottom
F3=Exit          F11=Remove a message      F12=Cancel
F13=Remove all   F16=Remove all except unanswered  F24=More keys
* - Work with problem allowed for message.

```

Figure 396. Display Messages for system operator message queue

You can obtain more detailed information for any of these messages by pressing the F1 (Help) key while the cursor is over a message. The messages are explained in greater detail in 22.3.3.1, “AS/400 messages” on page 739.

22.3.3.1 AS/400 messages

Message objects on the AS/400 system are stored in an object called a *message file* (*MSGF). They can be viewed using the AS/400 system Display Message Description (DSPMSGD) command. The default message file searched is QCPFMSG, which contains normal OS/400 messages (for example, CPF, CPI, and so on). When pressing the help function key (F1) while the cursor is over a message from a display such as DSPLOG or DSPJOBLOG, or while viewing a spooled file of a job log, the following elements are present in every message. Note the following points when calling for support:

- **Message ID:** A seven-character unique ID for the message. The first three characters form a prefix that identifies the origin of the message. Some of the most common prefixes are:
 - **MCH:** The message originated from Licensed Internal Code (LIC).
 - **CPx:** The message originated from the operating system (OS/400). The last character of the prefix can vary. For example, you can see such prefixes as CPI, CPD, or CPF.
 - **SQL:** The message originated from SQL run-time.
- **Message Type:** The type of the message. There are several message types, and they convey what type of situation has occurred. Some of the more common ones to see are:

- **Completion:** A message that conveys completion status of work.
 - **Diagnostic:** A message that indicates errors in a system function, errors in an application, or errors in input data.
 - **Escape:** A message that describes a condition for which a program must end abnormally. This is probably the most important one since it indicates an error situation, while other types convey information about events that have occurred and may not necessarily be an error condition.
 - **Information:** A message that provides general non error-related information.
 - **Inquiry:** A message that conveys information but also asks for a reply.
 - **Notify:** A message that describes a condition for which a program requires corrective action or a reply.
 - **Reply:** A message that is a response to a received inquiry or notify message.
 - **Request:** A message that contains a command for processing by a request processor, such as a command entry.
 - **Sender Copy:** A copy of an inquiry or notify message that is kept in the sender's message queue.
- **Severity:** The severity of the condition being reported and is a two-digit number that can be from 00 to 99, where 00 is the lowest severity and 99 is the highest severity. Typically, a severity level of 40 or above is a critical condition that needs attention. Some of the more common ones to see are:
 - **00 (Informational):** For informational purposes only. No reply is needed. The message can indicate that a function is in progress or that a function has completed successfully.
 - **10 (Warning):** A potential error condition exists. The program may have taken a default, such as supplying missing data. The results of the operation are assumed to be successful.
 - **20 (Error):** An error has been found, but it is one for which automatic recovery procedures probably were applied. Processing has continued. A default may have been taken to replace the wrong data. The results of the operation may not be correct. The function may not have completed. For example, some items in a list ran correctly, while other items did not.
 - **30 (Severe error):** The error found is too severe for automatic recovery procedures and no defaults are possible. If the error was in the source data, the entire data record was skipped. If the error occurred during a program, it leads to an abnormal end of program (severity 40). The results of the operation are not correct.
 - **40 (Severe error):** Abnormal end of program or function. The operation has ended, possibly because the program was not able to handle data that was not correct or because the user canceled it.
 - **50 (Abnormal end of job or program):** The job was not started or failed to start, a job-level function may not have been done as required, or the job may have been canceled.
 - **60 (System status):** Issued only to the system operator message queue. It gives either the status of or a warning about a device, a subsystem, or the system.

- **70 (Device integrity):** Issued only to the system operator message queue, indicating that a device is not working correctly or is in some way no longer operational.
 - **80 (System alert and user messages):** A condition exists that, although not severe enough to stop the system now, can become more severe unless preventive measures are taken.
 - **90 (System integrity):** Issued only to the system operator message queue. Describes a condition where either a subsystem or system cannot operate.
 - **99 (Action):** Some manual action is required, such as entering a reply or changing printer forms.
- **Date and Time:** The date and time the message was sent.
 - **From Program:** The name of the program that sent the message. The name can be in one of three formats:
 - The name starts with a # character. This means the sending program was in Licensed Internal Code.
 - The name consists of only library and program names. For example, OS/400 program names usually start with a Q and are in library QSYS. This is an old style program referred to as an Original Program Model (OPM) program. A program with a name of this type is not an ILE C program.
 - The name consists of four parts: library, program, module, and procedure names. This format is used for ILE programs. For ILE C, the procedure name is the C function name. When the name is in this form, the library and program names are probably on the initial line of the message with the module and procedure names following in the next lines.
 - **From Program Instruction Number:** The point in the sending program from where the message was sent. If this field contains a number, the sending program is not an ILE program. It is an OPM program and the number represents an MI instruction number. If the field contains the value *STMT, the program is an ILE program. This key value is used to signify that the number is not given on this line but on a following line. It also indicates that the number is a statement number relative to the source of the From Procedure that is also identified on a following line.
 - **To Program:** The name of the program to which the message was sent. Again, this name can be in one of the formats previously explained for the From Program. Additionally, the program name field can have the special value *EXT, which means the message was sent to the interactive session rather than a program.
 - **To Program Instruction Number:** The point in the program where the To Program was stopped when the message was sent. This field can contain a number or the special value *STMT, the same as previously described for the From Program Instruction Number.
 - **Message:** A text description of the condition being reported.
 - **Cause:** A likely cause of the condition.
 - **Recovery:** What actions should be taken, if any. If an answer is required, a list of possible choices is shown.

Use the F1 key to look at message details from a display such as DSPLOG or DSPJOBLOG. Figure 397 shows an example of the information you may see.

```
Additional Message Information

Message ID . . . . . : CPF0A51      Severity . . . . . : 00
Message type . . . . . : Information
Date sent . . . . . : 09/08/98      Time sent . . . . . : 00:02:40

Message . . . . . : Performance monitor 110194/QPGMR/QPFRMON started at
                   00:02:40 09/08/98.
Cause . . . . . : The job was started from job queue Q1PSCHQ2 in QMPGLIB.
                   The performance monitor was submitted to the job queue by
                   110193/QPM400/Q1PPMSUB. The data will be put into the performance data
                   collection database files located in library QMPGDATA with member name
                   Q982510002.

                                                                    Bottom

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details  F12=Cancel
F21=Select assistance level
```

Figure 397. AS/400 system message display: F1 key

To obtain additional information such as the from and to programs, press the F9 key. Figure 398 shows an example of the additional information you may see.

```
Display Message Details

Message ID . . . . . : CPF0A51      Severity . . . . . : 00
Date sent . . . . . : 09/08/98      Time sent . . . . . : 00:02:40
Message type . . . . . : Information
CCSID . . . . . : 65535

From job . . . . . : QPFRMON
  User . . . . . : QPGMR
  Number . . . . . : 110194

From program . . . . . : QPMMINIT

To message queue . . . . . : QSYSOPR
  Library . . . . . : QSYS

                                                                    Bottom

Press Enter to continue.

F1=Help  F3=Exit  F12=Cancel
```

Figure 398. AS/400 system message display: F9 key

When looking at a spooled file or printed version of a job log, Figure 399 shows an example of the information you may see in the 132 character mode.

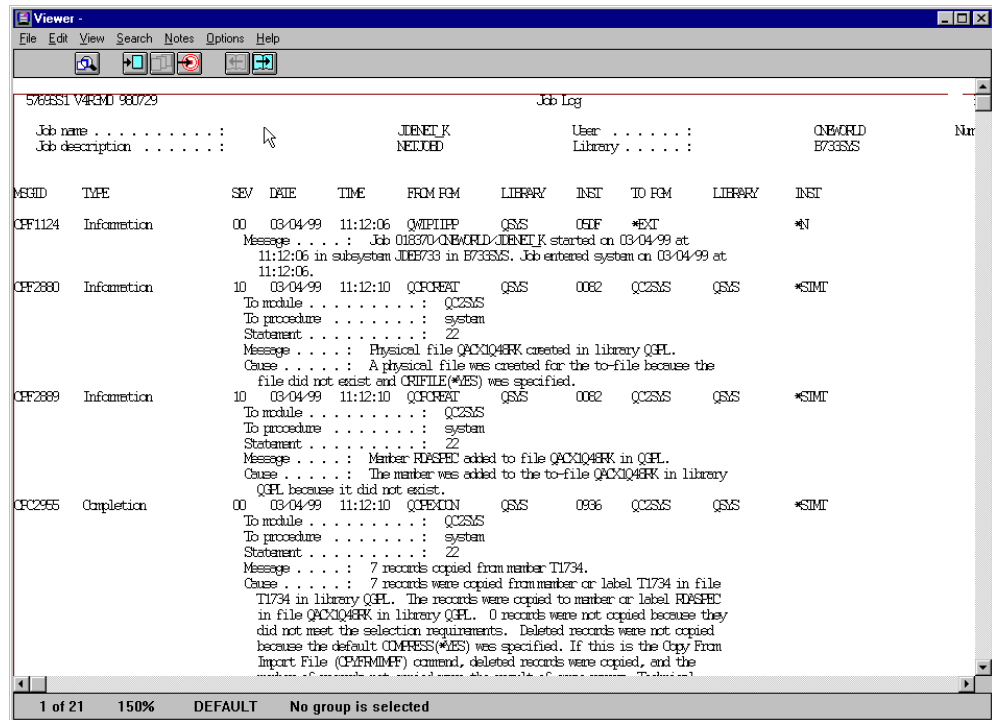


Figure 399. QPJOBLOG from Operations Navigator

If you are not using a display in the 132 character mode, use the F19 and F20 keys to move the display left and right to view the truncated characters. We recommend that you view job logs in the 132 character mode.

When debugging, the escape-type messages are important since they report error conditions encountered by OneWorld while they were running. However, some escape conditions are anticipated by the program, and the condition is handled so the job can continue. If an escape message is returned to the program, it must be anticipated and managed by the program, or the OneWorld job is terminated abnormally by the operating system. If you are looking at the job log for an active job and there is an escape message that is not followed by a termination message, you can be certain that the program managed the error condition for it to continue.

It is also important, when debugging, to find out if messages in the job log are related to the problem seen by the end user. If you see an error outside of a job log, sometimes you can associate it with a message in the job log. For example, if you see SQL error "-913" occurred..., you can search for the message SQL0913 in a job log. You need to verify if the time stamp in the job log matches the time stamp in which the error occurred (a few seconds of tolerance are allowed) to make sure this is the same event. In any case, check to see what messages were sent prior to this message. Look for escape-type messages around the time of the error.

If you are reviewing a job log in a spooled file, you can determine if an escape message was handled by looking at the messages that follow it. If you see an escape-type message and it is followed in the job log by another escape message with an ID of CPF9999 or CEE9901, the first escape message was not

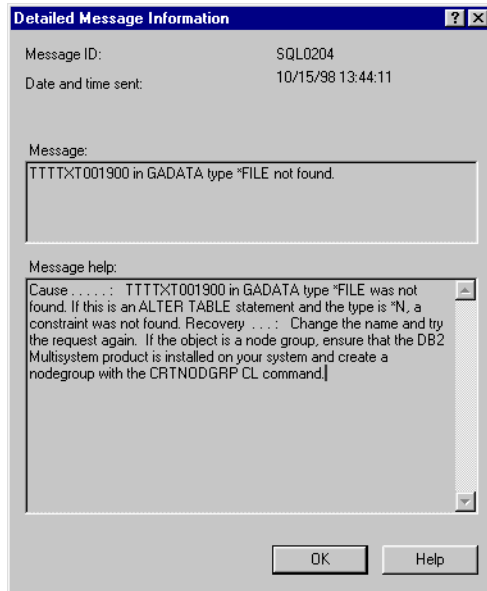


Figure 401. AS/400 Operations Navigator: Detailed message display

22.3.3.2 OneWorld client error messages

Always record the PC application error message, the Client Access ODBC error message, and the SQL State.

The PC application provides an error message. Include the exact text and error numbers in the error message. When a printer is unavailable, the image of the error message can be captured by pressing the Print Screen key. Print Screen copies a bitmap of the current screen to the clipboard. Use your preferred application to access and store the error bitmap in a folder for future reference.

22.3.3.3 Client Access ODBC error messages

The Microsoft ODBC specification defines a standard, layered error handling protocol. Error messages include the identity of the error source, the actual error text, the ODBC SQLSTATE, and the return code. ODBC error messages have the following format:

[vendor] [ODBC-component] [data-source] error-message text

The prefixes in brackets identify the vendor issuing the error, the ODBC component involved, and the data source (DBMS). Client Access ODBC uses the following format to identify errors whose source is the Client Access ODBC driver or DB2 UDB for AS/400:

[IBM] [Client Access ODBC Driver (32-bit)] [DB2/400 SQL]

Table 60 shows prefixes used in Client Access ODBC driver error messages.

Table 60. Prefixes used in Client Access ODBC driver error messages

Entry	Definition
[IBM]	Identifies IBM as the vendor issuing the error. If this value is something other than IBM, the failure did not occur in IBM code. An error starting with [Microsoft] [ODBCxxx DLL] implies the failure was at the MS ODBC driver manager level.

Entry	Definition
[Client Access ODBC Driver (32-bit)]	Identifies the component reporting the error. When the Client Access ODBC driver reports the error and the [DB2/400] prefix does not follow, the error occurred on the PC. The request was never sent to the AS/400 system.
[DB2/400]	Identifies the AS/400 system as the data source. This field, when included, indicates that the failure originated on the AS/400 system. The AS/400 job log may contain additional information for these types of errors.

Following the prefix labels in the error message, there is an error message identifier and the actual error text. If DB2 UDB for AS/400 returns an error message, the identifier is generally a PWS*, SQL*, or MCH* error. PWS* errors are issued by the database host server, while SQL* errors are issued by the DB2 UDB for AS/400 SQL programs. MCH* errors occur in either location.

Client Access service packs

Updates to the Client Access ODBC driver are included in the Client Access service packs. Verify that the latest Client Access service pack is applied on the PC. Service packs can be ordered through ECS or downloaded from the Client Access home page on the Web: <http://www.as400.ibm.com/clientaccess/>

22.3.4 Traces

This section describes how to use different types of traces.

22.3.4.1 AS/400 communications trace

The OS/400 communications trace facility is an important tool for isolating both performance problems and error conditions. It traces and formats any communications type that has a line description (for example, Token-Ring, Ethernet, or SDLC).

AS/400 communications traces are helpful in diagnosing where a performance delay is occurring. Use the time stamp and the eye-catcher fields to measure how long the AS/400 system takes to process a request.

Communications traces also capture the data stream being passed between Client Access ODBC and the AS/400 database host server. IBM Support can break down this data stream to see what is getting passed between the PC and the AS/400 system, including the SQL conversation and SQL statements, data definitions, and errors.

Sometimes, when trying to track down an AS/400 system problem, it may be determined that the problem is not with the system. It may possibly be somewhere between the AS/400 system (server) and the end user (client). OS/400 provides a communications trace facility that can show detailed information exchanged between the AS/400 system and a remote station or control unit.

Note

A large volume of information is gathered, even for a rather short period of time. Therefore, we recommend that the trace be activated for the shortest possible period.

Users with the correct levels of authority can invoke the function either through System Service Tools (SST), which is started by the AS/400 system Start System Service Tools (STRSST) command or by using the AS/400 system Start Communications Trace (STRCMNTRC) command. The trace should be ended from SST or by using the AS/400 system End Communications Trace (ENDCMNTRC) command. After ending the trace, the output can be formatted in many different ways. For example, when using TCP/IP, the output can be narrowed down to a specific IP address. This is helpful when looking for problems between the AS/400 system and a specific device. Use the following sequence to perform a communications trace using SST:

1. Issue the STRSST command. The display shown in Figure 402 appears.

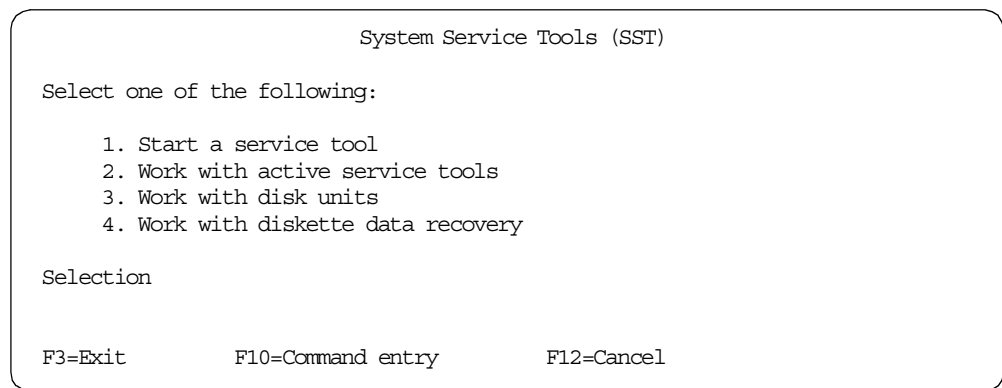


Figure 402. System Service Tools (SST) display: Main menu

2. Enter option 1, Start a service tool. The display shown in Figure 403 on page 748 appears.

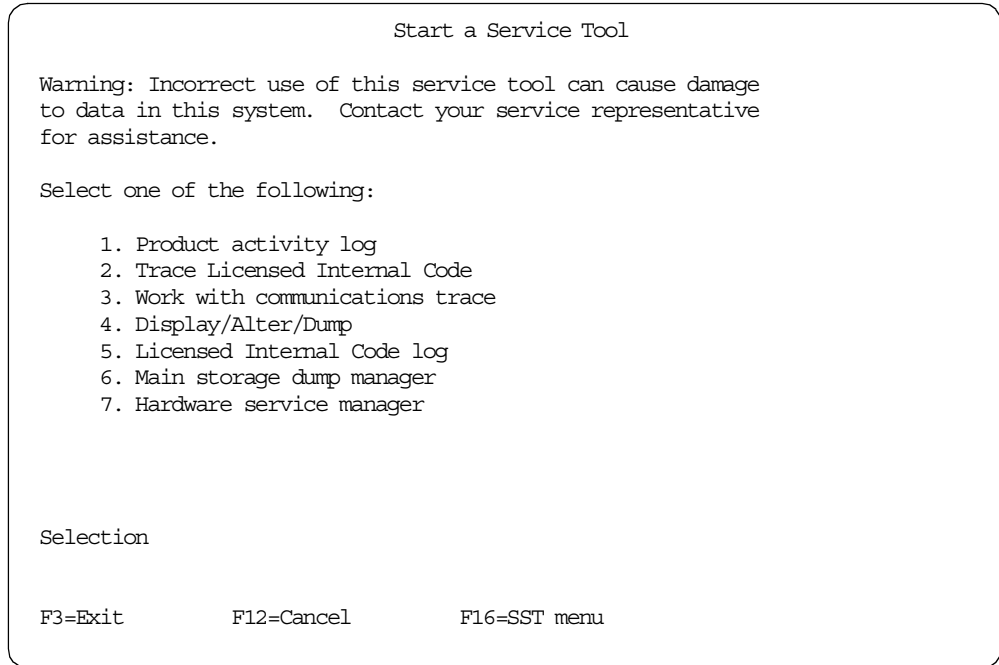


Figure 403. System Service Tools (SST) display: Option 1

3. Enter option 3, Work with communications trace. The display shown in Figure 404 appears.

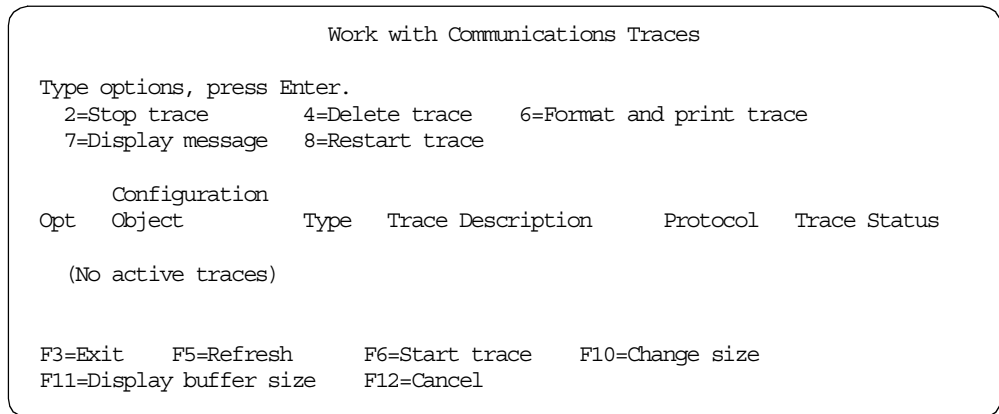


Figure 404. System Service Tools (SST) display: Option 3

4. Press the F6 key to start a trace. The display in Figure 405 appears.


```

                                Start Trace

Type choices, press Enter.

Configuration object . . . . . Line_Name

Type . . . . . 1      1=Line, 2=Network interface
                                3=Network server

Trace description . . . . . Some Meaningful Text

Buffer size . . . . . 9      1=128K, 2=256K, 3=2M, 4=4M
                                5=6M, 6=8M, 7=16M, 8=32M
                                9=64M

Stop on buffer full . . . . . N      Y=Yes, N=No

Data direction . . . . . 3      1=Sent, 2=Received, 3=Both

Number of bytes to trace:
  Beginning bytes . . . . . *CALC Value, *CALC
  Ending bytes . . . . . *CALC Value, *CALC

F3=Exit  F5=Refresh  F12=Cancel

```

Figure 405. System Service Tools (SST) display: F6 key

5. Enter the line description name with a meaningful text description and the largest buffer size allowed. Complete the Stop on buffer full parameter. Press the Enter key, and the trace goes active (Figure 406). Use the F5 key to refresh the display.

```

                                Work with Communications Traces

Type options, press Enter.
  2=Stop trace      4=Delete trace      6=Format and print trace
  7=Display message 8=Restart trace

Configuration
Opt  Object      Type  Trace Description  Protocol  Trace Status
    LINETRN     LINE  SOME MEANINGFUL TEXT  TRN      ACTIVE

F3=Exit  F5=Refresh  F6=Start trace  F10=Change size
F11=Display buffer size  F12=Cancel

```

Figure 406. System Service Tools (SST) display: Option 3

6. Recreate the problem and stop the trace by entering menu option 2, Stop trace. Remember that it does not take long for the buffer to fill up. After the trace stops, enter option 6, Format and print trace. The display in Figure 407 on page 750 appears.

```

                                Format Trace Data

Configuration object . . . . . : LINETRNL
Type . . . . . : LINE

Type choices, press Enter.

Controller . . . . . *ALL          *ALL, name

Data representation . . . . . 3      1=ASCII, 2=EBCDIC, 3=*CALC

Format RR, RNR commands . . . . . N      Y=Yes, N=No
Format Broadcast data . . . . . N      Y=Yes, N=No
Format MAC or SMT data only . . . . . N      Y=Yes, N=No
Format UI data only . . . . . N      Y=Yes, N=No
Format SNA data only . . . . . N      Y=Yes, N=No
Format TCP/IP data only . . . . . Y      Y=Yes, N=No
Format IPX data only . . . . . N      Y=Yes, N=No

F3=Exit      F5=Refresh      F12=Cancel

```

Figure 407. System Service Tools (SST) display: Option 6 (Part 1 of 2)

- If you want TCP/IP information, enter a `y` next to the Format TCP/IP data only parameter. Press the Enter key. The display in shown in Figure 408 appears. If desired, enter specific IP addresses to narrow down the report.

```

                                Select IP addresses to format

Configuration object . . . . . : LINETRNL
Type . . . . . : LINE

Type choices, press Enter.

IP address . . . . . *ALL          *ALL, address

IP address . . . . . *ALL          *ALL, address

F3=Exit      F5=Refresh      F12=Cancel

```

Figure 408. System Service Tools (SST) display: Option 6 (Part 2 of 2)

- When you are finished, the system creates a spooled file called QPCSMPT with detailed information regarding communications activity during the trace time that is formatted the way you specified. You may need someone familiar with viewing these files to determine what looks normal.

As an alternative, use the `STRCMNTRC` and `ENDCMNTRC` commands. Figure 409 and Figure 410 show examples of using these commands.

```

                                Start Communications Trace (STRCMNTRC)

Type choices, press Enter.

Configuration object . . . . . > LINE_NAME      Name
Type . . . . . > *LIN                        *LIN, *NWI, *NWS
Buffer size . . . . . > *MAX                  *MIN, *MAX, 128K, 256K, 2M...
Data direction . . . . . *BOTH                 *SND, *RCV, *BOTH
Trace full . . . . . > *STOPTRC              *WRAP, *STOPTRC
Number of user bytes to trace:
  Beginning bytes . . . . . *CALC              Number, *CALC
  Ending bytes . . . . .                      Number, *CALC
Communications trace options . . *ALLDTA      *ALLDTA, *RMICTL, *RMIMAC...
Trace description . . . . . Some Meaningful Text

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 409. Start Communications Trace (STRCMNTRC) display

```

                                End Communications Trace (ENDCMNTRC)

Type choices, press Enter.

Configuration object . . . . . Line_Name      Name
Type . . . . . *LIN                        *LIN, *NWI, *NWS

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 410. End Communications Trace (ENDCMNTRC) display

22.3.4.2 AS/400 job trace

When a detailed analysis of a specific job is needed, OS/400 also has a job trace facility. It can show such data as program, module, or procedure call and return sequences. The information present in the printed report includes:

- **Time stamp:** Based on the AS/400 clock or system time
- **Library/program:** Indicates the program and library involved
- **Resource utilization by program:** Measures:
 - CPU utilization
 - DB/non-DB reads
- **OS/400 program “debug” information unique to the OS/400 component:** This can be data being passed or messages being sent between modules or programs.

Job trace function without Performance Tools/400

Follow these steps to perform a job trace:

1. Issue the AS/400 system Start Service Job (STRSRVJOB) command. The display shown in Figure 411 on page 752 appears.

```

                                Start Service Job (STRSRVJOB)

Type choices, press Enter.

Job name . . . . . > QZDASOINIT      Name
User . . . . . > QUSER              Name
Number . . . . . > 094199           000000-999999

                                                Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display      F24=More keys

```

Figure 411. Start Service Job (STRSRVJOB) display

This is done against the job about which you want to gather information. You need to know the qualified job name (for example, job name, user, or number). This can be found by using such AS/400 system commands as WRKACTJOB or WRKJOB.

- Issue the AS/400 system Start DEBUG (STRDBG) command. The display in Figure 412 appears.

```

                                Start Debug (STRDBG)

Type choices, press Enter.

Program . . . . . *NONE              Name, *NONE
Library . . . . .                               Name, *LIBL, *CURLIB
      + for more values

Default program . . . . . *PGM        Name, *PGM, *NONE
Maximum trace statements . . . . . 200      Number
Trace full . . . . . *STOPTRC        *STOPTRC, *WRAP
Update production files . . . . . *YES    *NO, *YES
OPM source level debug . . . . . *YES    *NO, *YES
Service program . . . . . *NONE        Name, *NONE
Library . . . . .                               Name, *LIBL, *CURLIB
      + for more values

                                                More...
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display      F24=More keys

```

Figure 412. Start Debug (STRDBG) display

This places the job into debug mode. Sometimes it places more messages into the job log. Make sure that Trace Full is set to *STOPTRC, Update production files is set to *YES, and OPM source level debug is set to *YES.

- Issue the AS/400 system Trace Job (TRCJOB) command. The display shown in Figure 413 appears.

```

                                Trace Job (TRCJOB)

Type choices, press Enter.

Trace option setting . . . . . *ON          *ON, *OFF, *END
Trace type . . . . . *ALL          *ALL, *FLOW, *DATA
Maximum storage to use . . . . . 16000      1-16000 K
Trace full . . . . . *STOPTRC      *WRAP, *STOPTRC
Program to call before trace . . *NONE      Name, *NONE
  Library . . . . . *NONE          Name, *LIBL, *CURLIB
Select procedures to trace:
  Program . . . . . *ALL          Name, *ALL, *NONE
  Library . . . . . *NONE          Name, *LIBL, *CURLIB
  Type . . . . . *PGM, *SRVPGM
                                + for more values

                                                                Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 413. Trace Job (TRCJOB) display

This starts the tracing of the job. Make sure that Trace option setting is set to *ON, Trace type is set to *ALL, Maximum storage to use is set to 16000, and Trace full is set to *STOPTRC.

4. Issue the AS/400 command Change Job (CHGJOB) command. The display shown in Figure 414 appears.

```

                                Change Job (CHGJOB)

Type choices, press Enter.

Job name . . . . . > ONEWORLD      Name, *
User . . . . . > JDE              Name
Number . . . . . > 104957         000000-999999
...
...
...
Message logging:
  Level . . . . . 4                0-4, *SAME
  Severity . . . . . 00            0-99, *SAME
  Text . . . . . *SECLVL          *SAME, *MSG, *SECLVL, *NOLIST

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 414. Change Job (CHGJOB) display

This forces the job to produce a job log if it is not already set to do so. Make sure that Message logging is set to 4, 00, and *SECLVL.

5. Recreate the problem or error.

6. Issue the following commands to end the trace of the job, to end debug mode, and to end the service job:

- TRCJOB SET (*OFF)
- ENDDBG
- ENDSRVJOB

7. When done, the system creates a spooled file called QPSRVTRC with detailed information regarding job activity during the trace time. You may need someone familiar with viewing these files to determine what looks normal. Figure 415 shows an example of the information you may see.

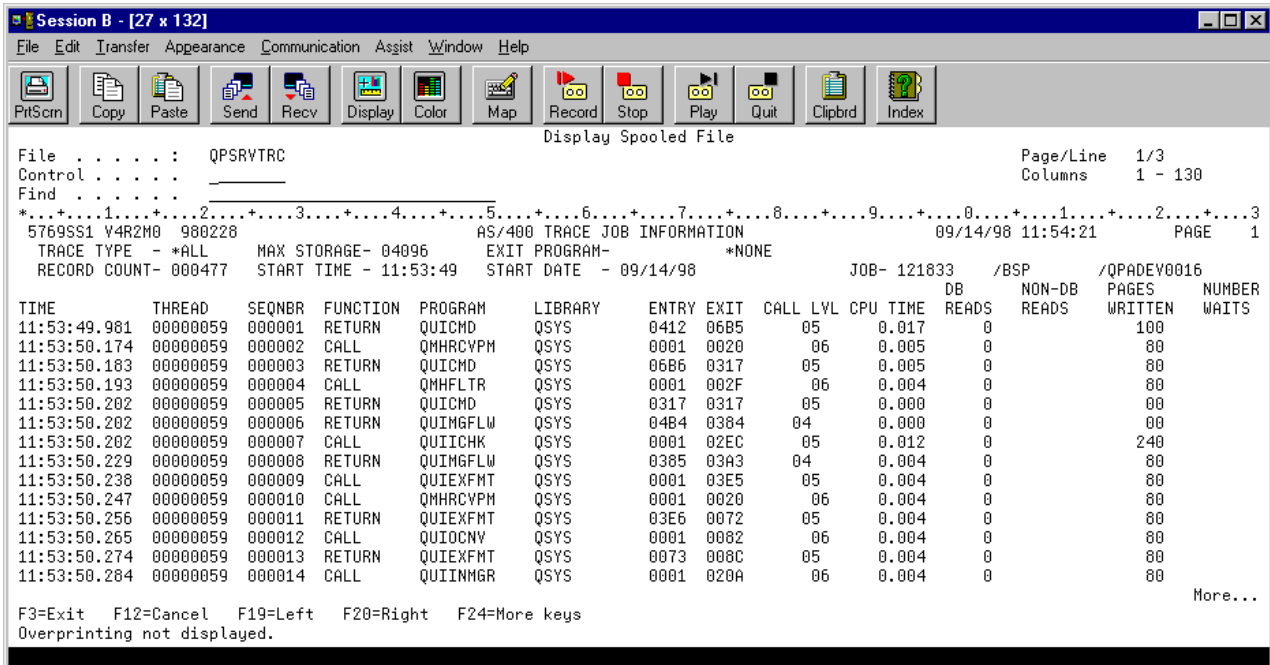


Figure 415. QPSRTRC job trace spooled file display

Job trace function with Performance Tools/400

An alternative approach is to use the start job trace function available within Performance Tools for AS/400 (5769-PT1). The quick steps to perform this are:

1. Type GO PERFORM to go to the Performance Tools for AS/400 main menu.
2. Enter option 5, Performance utilities.
3. Enter option 1, Work with job traces.
4. Enter option 1, Start job trace.
5. Recreate the problem or error.
6. Select option 2, Stop job trace.
7. Enter option 3, Print job trace reports.

22.3.5 OneWorld traces

OneWorld traces are track within OneWorld logs, described in 22.3.2.1, “OneWorld logs” on page 734.

22.3.5.1 Client Access ODBC traces

An ODBC trace captures ODBC connection information, ODBC API calls, SQL statements, and ODBC error messages. An ODBC trace is useful in both error

diagnostics and performance tuning. The trace occurs at a level between the application and the ODBC driver so it is helpful in showing what is passed to the ODBC driver from the application. ODBC 3.0 includes its own trace utility.

To perform an ODBC trace, follow this process:

1. Click **Start->Programs->IBM AS/400 Client Access->ODBC Administrator**.
2. Select the **Tracing** tab, and click **Start Tracing Now** (Figure 416).

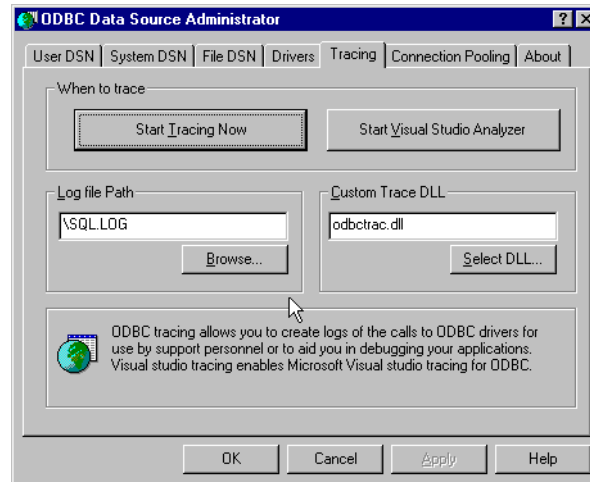


Figure 416. Start ODBC tracing

3. Recreate the error.
4. Select the **Tracing** tab again, and click **Stop Tracing Now** to end the trace. Do not leave the trace running since it will have a severe performance impact on ODBC jobs (Figure 417).

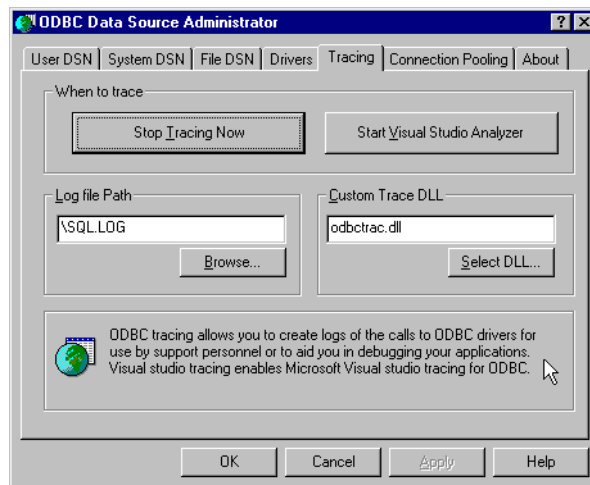


Figure 417. Stop ODBC trace

22.4 Installation verification using PORTTEST

The OneWorld AS/400 components required to run OneWorld on the AS/400 platform are loaded during the Enterprise Server Installation. The first step in verifying that the AS/400 Enterprise Server is installed correctly is to run the

PORTTEST command. PORTTEST is a OneWorld command that, when executed, verifies the basic components of the installation by loading an environment and performing database table access through the J.D. Edwards middleware. Use the following steps to run PORTTEST and verify its results. If you encounter problems, refer to 22.4.1, “PORTTEST problem resolution” on page 758.

1. Sign on to the AS/400 Enterprise Server as user `ONEWORLD`.
2. Start the OneWorld services by issuing the `STRNET` command.

If Security Server is turned on, PORTTEST uses the Security Kernel to verify the USER and Password against the F98OWSEC table. In the AS/400 INI the [SECURITY] section has an entry ‘SecurityServer=<Machine_Name>’. This mentioned <Machine_Name> must be a valid OneWorld Enterprise Server with OneWorld host code loaded on it. In this scenario, OneWorld services must be started to process this request.

If Security Server is turned off, PORTTEST can be executed without starting OneWorld services. Place a semi-colon (;) in front of the ‘SecurityServer=<Machine_Name>’ item in the [Security] section of the AS/400 INI to turn Security Server off on the AS/400 system.

3. Verify that you have a JDEB7332 subsystem running on your AS/400 system with an active NETWORK and SENTINEL job (Figure 418).

```

Work with Active Jobs
ASM23
02/29/00 14:54:03
CPU %: .0 Elapsed time: 00:00:00 Active jobs: 163

Type options, press Enter.
 2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
 8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
JDEB733 QSYS SBS .0 DEQW
NETWORK ONEWORLD BCH .0 PGM-JDENET_N SELW
SENTINEL ONEWORLD ASJ .0 PGM-MONITOR SIGW
More...

Parameters or command
====>
F3=Exit F5=Refresh F10=Restart statistics F11=Display elapsed data
F12=Cancel F23=More options F24=More keys

```

Figure 418. WRKACTJOB SBS (JDEB733)

- The Network JDENET_N job is used to process all incoming OneWorld communication messages.
 - The SENTINEL job monitors the number of JDB_n programs that are available and in use. The JDB_n programs are used to ensure commitment control is correctly scoped.
4. Run the `PORTTEST` command by prompting for values (F4). See Figure 419.


```

                                RUN PORTTEST (PORTTEST)

Type choices, press Enter.

User Profile . . . . . JDE           Name
User Password . . . . . JDE         Character value
Environment . . . . . PRD733       Name

                                                                 Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 419. Run PORTTEST command

The successful execution of PORTTEST produces results similar to those listed in Figure 420. On this display, additional information scrolls by. The display in Figure 420 represents the end of the PORTTEST output.

```

Calling DataDictionary Validation function,...
Data Dictionary Validation Succeed for "CO 00001".

Closing table F0902,...
Closed table F0902.

Freeing user JDE,...
Freed user JDE successfully.

Cleaning up environment PRD733,...
Cleaned up environment PRD733 successfully.

Congratulations! Porttest completed successfully.

All Done!
Bye!

===>

F3=Exit F4=End of File F6=Print F9=Retrieve F17=Top
F18=Bottom F19=Left F20=Right F21=User Window

```

Figure 420. Successful PORTTEST output

If PORTTEST does not run successfully, you receive output on your display similar to the example in Figure 421 on page 758.

```

Running porttest for JDE on PRODB733 with password JDE,...

Initializing Environment PRODB733,...
Initialization of environment failed - rcode = 0
    See host services check-list for details.

Porttest failed.
    See the host services check-list for details.

All Done!
Bye!

====>

F3=Exit F4=End of File F6=Print F9=Retrieve F17=Top
F18=Bottom F19=Left F20=Right F21=User Window

```

Figure 421. Unsuccessful PORTTEST

22.4.1 PORTTEST problem resolution

To verify the AS/400 OneWorld installation and research PORTTEST failures, perform the following steps:

1. Ensure that the host code installation completed successfully. To do this, complete the checklist shown in Table 61.

Table 61. AS/400 host code checklist

X	Verification action	Step to resolve
	Enter the <code>WRKLIB B7332SYS</code> command to verify that you have a B7332SYS library. B7332SYS should contain approximately 129 objects.	B7332SYS is restored during the Host Code installation. Check the job log for errors and rerun.
	Enter the <code>WRKLIB B733MAP</code> command to verify that you have a B733MAP library that contains approximately 38 objects.	B733MAP is created and populated during the Planner and Workbench portions of the installation. Verify and rerun it.
	Enter the <code>WRKLNK</code> command. Verify that your pathcode directories exist in the following format: <code>/prodb733/specfile/*.*</code> . You should have an IFS directory structure for each pathcode you defined in your plan.	The host code installation creates the IFS directory structure. Check the job log for errors, correct, and rerun. The pathcodes to install are determined by the LIBRARY file that resides on the Deployment Server in the <code>B733\Hosts\As400\AS400name\library</code> directory. This text file is sent by FTP to the AS/400 system during LODRUN.

X	Verification action	Step to resolve
	Use the <code>WRKLIB</code> command to verify your pathcode libraries. You should have a library for each pathcode you defined in your plan, for example <code>PRODB733</code> . This library should contain approximately 29 objects.	The host code installation creates the pathcode libraries. Check the job log for errors, correct, and rerun. The pathcodes to install are determined by the <code>LIBRARY</code> file that resides on the Deployment Server in the following directory: <code>B733\Hosts\As400\AS400name\library</code> . This text file is sent by FTP to the AS/400 system during <code>LODRUN</code> .
	Verify that the <code>INI</code> file in the <code>B7332SYS</code> library contains valid environments, pathcodes, and machines names.	The <code>INI</code> file is restored during the host code installation execution of <code>LODRUN</code> . Check for errors and rerun.

Note: The next step assumes that debugging is turned on as configured in the `jde.ini` file.

2. Examine the JDE and `jddebug` logs on the AS/400 system. Running `PORTTEST` creates a `JDE_nnnnnn.log` and `jddebug_nnnnnn.log`, where `nnnnnn` represents the job number for the interactive job to which you are signed on.

To find your job number, enter the `DSPJOB` command, and note the job number associated with your interactive job, for example `053150`. The errors encountered by `PORTTEST` are found in the `JDE_053150.log` file and `jddebug_053150.log` in the log file directory of the IFS. When examining the `PORTTEST` logs, start with the `jde.log`, and review it from the top down. The `jde.log` contains any errors that were encountered. The `jddebug.log` contains a running history of the steps executed by the process. Figure 422 shows an example of a `PORTTEST jde.log` where errors were encountered.

```

13316 Wed Mar 10 15:13:50 1999 JDEIPC_D/C/IPCPUB1890
process 13316 <XB733SYS/PORTTEST> registered in entry 2

13316 Wed Mar 10 15:14:00 1999 JDEKRNL_D/C/JDB_OMP1467
JDB9900244 - No default record exist for tables for environment PRODB733 in OCM
.

13316 Wed Mar 10 15:14:00 1999 JDEKRNL_D/C/JDB_CACH1024
JDB9900009 - Failed to load ObjectMap Cache

13316 Wed Mar 10 15:14:00 1999 JDEKRNL_D/C/JDB_CTL2404
JDB1100016 - Failed to load Environment - PRODB733

13316 Wed Mar 10 15:14:00 1999 JDEIPC_D/C/IPCPUB2148
process 13316 unregistered in entry 2

```

Figure 422. `PORTTEST jde.log`

3. The initial execution of `PORTTEST` after a new installation, upgrade, or update creates additional objects on the AS/400 system. If `PORTTEST` fails, check to see if these objects were created. This provides information about where the problem may have occurred. These objects are:

- **JDB_n programs** in the B7332SYS library. These programs are used to scope commitment control boundaries. The JDE_PGMCTL *USRSPC object in library B7332SYS controls whether these need to be created.
- **GLBLTBL.XDB** and **GBLTBL.DDB stream file objects** are created in the IFS directory of the pathcode over which you ran PORTTEST. The GLBLTBL files contain file layout details used by OneWorld.

22.5 Tips and techniques

This section contains useful tips and techniques for problem identification and solution.

22.5.1 General troubleshooting hints

This section offers several valuable hints you can use for general problem solving:

- Narrow down the definition of any problem that you may have, particularly when communicating the issue to someone else (for example, J.D. Edwards' Customer Support). For example, do not say “The UBE failed”. Instead, say how you know that the UBE failed: “The UBE status is E”; “The report had the wrong data”; and so on.
- When communicating an error message, be sure to include all parts of the error message exactly as they appear in the log file or on the screen. Parts of the message that may not seem important to you may actually hold the key as to why an error occurs. Also, distinguish between characters that may be misinterpreted (for example, the capital letter “O” and the number zero “0”). We highly recommend that you fax the error messages.
- As soon as you notice an error, examine the log files. Messages near the bottom of the log files may reveal the most important information about the cause of the error.
- Before you restart OneWorld on the server, either delete or move the JOB_XXXXXX (where XXXXXX is a number) members from the jde.log and jdedebug files.
- When you first try to run OneWorld, ensure you have logging on (see Chapter 12, “Setting Up the OneWorld AS/400 INI File” in the J.D. Edwards *OneWorld Server and Workstation Administration Guide*). Examine the jde.log and jdedebug files carefully.
- Carefully examine the AS/400 job logs and JDE jde.log files of the OneWorld jobs to help ensure that authorities and OCM are set correctly.
 - Look for messages in the jde.log files, such as JDB3100011 Failed to get location of table F983051 for environment PRODB732.
 - Look for messages in the AS/400 job logs similar to File F98306 not found in library PRODDTA.
- You may want to temporarily modify the job description of the ONEWORLD user profile to always write the AS/400 job log until you are comfortable that everything is set up correctly.

22.5.2 OneWorld UBEs work inconsistently

In some cases, you may find that a specific UBE works inconsistently between the OneWorld client workstation and the logic server. It is also possible for the UBE to work inconsistently between OneWorld environments by related pathcodes.

Ensure that the problem is not caused by inconsistent Object Configuration Mappings (OCM) between the “System - B733” and “<host> - Server Map” Data Sources locations. Once you are satisfied that this is not the case, there may be three contributing factors for this issue:

- It is possible that the code on the client machine differs from the code on the logic server.
 - a. From Object Librarian, check out the UBE to a client machine and test it. Determine if UBE still runs locally on the client. If so, erase the check out record.
 - b. From menu GH908, create a server package containing the UBE and any related dependencies. Transfer and deploy server package to logic server for single pathcode, for example, TESTB733. Test the UBE after the server package successfully deploys to the logic server. If the problem persists, go to the next bulleted item.
- It is possible that the global tables on the logic server contain bad information or have become corrupted. Sign on as user `ONEWORLD` to complete the following items:
 - a. End OneWorld services, using the `ENDNET` command.
 - b. Clear the inter-process-communications semaphores, shared memory, and message queues, using the `CLRIPC` command.
 - c. Remove the global table object links from a specific pathcode IFS directory, for example, TESTB733. Use the command:

```
RMVLNK(<pathcode>/specfile/glbltbl*)
```
 - d. These global table object links are recreated dynamically either through `PORTTEST` or the submission of the first UBE against the specified environment or through the pathcode, for example, TESTB733.
 - e. Start OneWorld services, using the `STRNET` command.
 - f. Test the UBE launched against the logic server. If the problem persists, go to the next bulleted item.
- On occasion, the AS/400 SQLPKGs used by OneWorld may become corrupted.
 - a. To ensure that interactive OneWorld sessions are not adversely affected, ask all users to sign-off of `ONEWORLD`.
 - b. End OneWorld services using the `ENDNET` command.
 - c. Clear the inter-process-communications semaphores, shared memory, and message queues using the `CLRIPC` command.
 - d. End host server, using the command:

```
ENDHOSTSVR *ALL
```
 - e. End the QSERVER subsystem to prevent users from signing on. Enter:

```
ENDSBS QSERVER *IMMED
```

- f. Delete all JDE and ONEWORLD created SQL packages. Use the command:

```
WRKOBJ *ALL/*ALL *SQLPKG
```

Enter option 8, Display Description, to identify the created by user if you are uncertain of the packages origin. Enter option 4, Delete, to delete all appropriate SQL packages.

Attention

Take care not to delete IBM supplied SQL packages, for example, QIWS/QZDAPKG or QSYS/QSQLPKG2.

- g. Start the QSERVER subsystem by using the command:

```
STRSBS QSERVER
```

- h. Start host server using the command:

```
STRHOSTSVR *ALL
```

Ignore the IPX failure messages.

- i. The SQL packages are re-created during normal OneWorld signon and application usage.
- j. Start OneWorld services using the STRNET command.

22.5.3 Debugging a Business Function (BSFN) on a logic server

A OneWorld Business Function or BSFN is created as a CMOD (or C module) object on an AS/400 system. These CMODs are contained in a SRVPGM, or service program, which is the equivalent of a *.DLL in a Windows NT environment (such as a workstation). The Parent DLL for a BSFN (found in Object Librarian) is the corresponding SRVPGM for this BSFN when used on an AS/400 system. For example, the B4200310 BSFN is contained in the CDIST.DLL on a Windows NT system, and in the CDIST SRVPGM on an AS/400 system. On an AS/400 system, the B4200310 is a CMOD.

22.5.3.1 BSFN calls from interactive applications

Complete the following tasks:

1. Verify the name of the pathcode library on the AS/400 system for the environment in which you will debug.
2. Backup (typically using the CRTDUPOBJ command) the CMOD and SRVPGM objects.
3. Your library list should include the following elements (use the EDTLIBL command to see it):
 - **Pathcode library:** Example of library names: PRODB732
 - **System library:** B7332SYS
4. Create the function using the CRTCMOD command. The parameters can be set as shown here:
 - **Module:** BSFN name, for example B4200310
 - **Library:** Pathcode library
 - **Source File:** SRVPGM, or DLL name (CFIN, for example)
 - **Library:** Pathcode library

Press F10 for further options.

- **Optimize:** *NONE when you are debugging
- **Debugging view:** *ALL when you are debugging
- **Target Release:** V4R3M0 or V4R4M0 for B7332

5. Update the SRVPGM using the `UPDSRVPGM` command as shown here:

- **Service program:** SRVPGM or DLL name. CDIST for example
- **Library:** Pathcode library
- **Module:** BSFN name, for example B4200310
- **Library:** Pathcode library

Press F10 for further options.

- **Creation options:** Use the following three options: *DUPVAR, *DUPPROC, *UNRSLVREF. The + allows you to add multiple values.

6. Set the connectTimeout parameter under the [JDENET] Section higher (for example, 3600) on the workstation jde.ini.

7. Find the JDENET process that your interactive session is using for running processes on the server (use the `WRKACTJOB` command). Since you may typically debug in an environment when few users are also logged in, you should be able to find the JDENET process by testing the program that you will debug. OneWorld tries to use the same JDENET process each time you perform this test. To find the correct process, follow these steps:

- a. Test the application.
- b. Set your current directory to JDEB7332 (the JDE directory for your release). To do so, enter the `WRKLNK` command and enter option 11 next to this directory.
- c. Use Display file (`DSPF`) command on the most recent logs (last in the list) in this directory. You should find the call to the function you are trying to debug in the log. Your function call may not show if it is an embedded call. You may have to find the function that calls your function, and so on. The job number that is part of the log name is your JDENET process.
- d. Enter option 5 from the `WRKACTJOB` menu for the JDENET process, and find the one that shares this job number (upper right corner).
- e. Record the Job, User, and Number from this process.

8. Start a service job. Use the `STRSRVJOB` command and the Job, User, and Number from the JDENET process.

9. Start debug. Use the `STRDBG` command for the JDENET_K process as the program. This brings you into the debug session in a Display Module Source screen.

10. Press F14 to see the programs that are in debug. Your function will not show. Select option 1 to add in a line for your SRVPGM in the pathcode library:

```
CFIN  PRODB733  *SRVPGM
```

11. Your BSFN should now appear in the list. Press F5 to display the source for this BSFN. Then, press F6 to set break points. Note that your OneWorld application hangs when the process enters Debug mode. Check your client access session to see if it stopped at a break point.

12. Run your test. When you are finished debugging, run the `ENDDBG` and `ENDSRVJOB` commands.

Notes and hints

- Never press F3 while a business function is executing in debug. This ends the program. Allow the business function to complete *prior* to ending debug. Failure to do so may result in data corruption, unreleased memory, and an unstable JDENET_K process.
- To add additional modules to your debug session, use the `DSPMODSRC` command and press F14. You will see the interface in Step 10 (for the interactive call) above.
- To display the value of a variable:
 - **Integer or enumerated:** `EVAL [variable name]` or `<F11>` cursor sensitive
 - **String** `EVAL *[string name]:S`, `EVAL *[string name]:SRX`
 - **Mathnumeric:** `EVAL *[variable name]`

Note: You do not need the asterisk (*) if the variable is not a pointer.
- In B7332, the AS/400 `jde.ini` is in your system library, file INI member JDE. In B732, it is in your control library, file INI with member JDE.

22.5.3.2 BSFN called from a UBE (batch process)

To debug a BSFN called from a UBE, perform this series of tasks:

1. Verify the name of the pathcode library on the AS/400 system for the environment in which you will debug.
2. Backup (typically using the `CRTDUPOBJ` command) the CMOD and SRVPGM objects.
3. Your library list should include (use the `EDTLIBL` command to see it):
 - **Pathcode library:** Example of library names: PRODB733
 - **System library:** B7332SYS
4. Create the function using the `CRTCMOD` command. The parameters can be set as shown here:
 - **Module:** BSFN name, for example B4200310
 - **Library:** Pathcode library
 - **Source File:** The SRVPGM, or DLL name (CFIN for example)
 - **Library:** Pathcode library

Press F10 for further options.

 - **Optimize:** *NONE when you are debugging
 - **Debugging view:** *ALL when you are debugging
 - **Target Release:** V4R3M0, or V4R4M0 for B7332
5. Update the SRVPGM using the `UPDSRVPGM` as follows:
 - **Service program:** SRVPGM or DLL name. CDIST for example
 - **Library:** Pathcode library
 - **Module:** BSFN name, for example B4200310
 - **Library:** Pathcode library

Press F10 for further options.

Use the following three options: `*DUPVAR`, `*DUPPROC`, `*UNRSLVREF`. The + allows you to add multiple values.

6. Submit the UBE. Find the job number (WRKJOBQ, you can find the jobq from Work With Servers in OneWorld).
7. Start a service job. Use the `STRSRVJOB` command and the Job (R014021), User (DS5598241) and Job Number (from 8 above).
8. Start debug. Use `STRDBG` command:
 - **Program:** PRINTUBE
 - **Update production files:** *Yes
9. The `STRDBG` command brings you to a Display Module Source screen. Press F14 to see the programs that are in debug (there should not be any yet). Select option **1** to add in a line for the `SRVPGM` menu, which contains your `BSFN`. For example, B9800190 is in `CRUNTIME`:
 - **Program:** CRUNTIME
 - **Library:** *LIBL
 - **Type:** *SRVPGM
10. Release the jobq (RLSJOBQ). This brings up a message indicating the serviced job is released from the job queue. Press F10 to access a command line to set break points.
11. From the command line, enter the `DSPMODSRC` command to return to the source code for your `BSFN`. Set break points by pressing F6. When you set all of the break points you want, press F12 until you arrive at the screen indicating that the serviced job is released, and press Return to start the job.
12. The system brings up your debug session, stopping at the first break point encountered. You can use the function keys available in debug to step through the program and so on.
13. When you are finished debugging and have returned to a command line, enter the `ENDDBG` and `ENDSRVJOB` commands.

Notes and hints

- Never press F3 while a business function is executing in debug. This ends the program. Allow the business function to complete *prior* to ending debug. Failure to do so may result in data corruption, unreleased memory, and an unstable `JDENET_K` process.
- To add additional modules to your debug session, use the `DSPMODSRC` command, and press F14. You will see the interface in step 10 (for the interactive call) above.
- To display the value of a variable:
 - **Integer or enumerated:** `EVAL [variable name] <F11>` cursor sensitive
 - **String** `EVAL *[string name]:S, EVAL *[string name]:SRX`
 - **Mathnumeric:** `EVAL *[variable name]`

Note: You do not need the asterisk (*) if the variable is not a pointer.
- In B7332, the `AS/400 jde.ini` is in your system library, file `INI` member `JDE`. In B732, it is in your control library, file `INI` with member `JDE`.

22.5.4 CL program to start debug in QZDASOINIT jobs

This is the source for a CL program to start debug on QZDASOINIT Jobs. The AS/400 Worldwide Technical Support Group found this program to be extremely helpful in supporting our OneWorld customers. You can call this program whatever you prefer but remember the name and library:

```
PGM PARM(&STATUS &REQUEST)

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* PROGRAM CALL PARAMETER DECLARATIONS                               */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
DCL VAR(&STATUS) TYPE(*CHAR) LEN(1) /* ACCEPT/REJECT INDICATOR */
DCL VAR(&REQUEST) TYPE(*CHAR) LEN(34) /* PARAMETER STRUCTURE */
STRDBG UPDPROD(*YES)

CHGVAR VAR(&STATUS) VALUE('1')
ENDPGM
```

To activate this program with QZDASOINIT host server jobs, perform these steps:

1. Compile the above described program.
2. Enter `WRKREGINF` command on AS/400 command line.
3. Page down until you see this entry:

```
QIBM_QZDA_INIT          ZDAI0100          *YES          Database Server - entry
```

4. Enter option 8 next to this entry.
5. Enter option 1.
6. Type in Program Name and the library (the name and library you used when the program was created). Press Enter.
7. On the AS/400 command line, run the following commands to reset the Prestart Job (PJ). This activates the exit program with the host server job.

```
ENDPJ SBS(QSERVER) PGM(QIWS/QZDASOINIT) OPTION(*IMMED)
STRPJ SBS(QSERVER) PGM(QIWS/QZDASOINIT)
```

22.5.5 Running OneWorld

While running OneWorld on the AS/400 system, if you notice a marked slowing of response times, check for:

- **System operator messages:** There may be messages issued from jobs to the QSYSOPR system operator. An example of this is when you run the database monitor (`STRDBMON` command) for some time. A “file full” condition may occur that results in all of the OneWorld jobs waiting for operator intervention with the status of MSGW.
- **Object locks:** Use the `WRKACTJOB` command to look at the OneWorld jobs and see if the job status shows LCKW for long periods of time. This indicates a conflict in accessing an object on the system. Option 12 from the Work with Job display shows the locks held by a job. Once you find the object that is causing the problem, issue the AS/400 system Work with Object Locks (`WRKOBJLCK`) command to work with the object lock requests in the system for it. Another place to look is in the job log for “resource busy” messages such as SQL0913.
- **Disk occupancy:** Use the AS/400 system Work with System Status (`WRKSYSSTS`) command to check the disk usage of the system ASP (ASP1). The AS/400 system Work with Disk Status (`WRKDSKSTS`) command can be used to look at the disk space usage of specific disk units or user ASPs (for example,

ASP2, which may contain journal receivers). Do not allow user ASPs to exceed their capacity and overflow into the system ASP. Extremely full disk units (or an ASP) can cause poor system performance.

- **Errors in logs of message queues:** Check OneWorld system logs, client logs, the AS/400 system, QHST system log, AS/400 system job logs, the AS/400 system QSYSOPR message queue, and AS/400 system user message queues to look for errors that can impact user response times. Chapter 21, “Performance management” on page 561, provides more information regarding system tuning for performance problems.

22.5.6 SQL packages full, corrupted, or not being used

If you suspect a problem, delete the SQL packages and rebuild them automatically. To do so, perform these tasks:

1. Locate a job log for the QZDASOINIT job or user that first encountered the failure. Enter the WRKSPLF User ID or a WRKOUTQ QEZJOBLOG command.
2. Look for messages in QSYSOPR or in the user message queue.
3. Check output queue QEZJOBLOG for job logs for that user, and search for messages in the job logs.
4. Verify whether the package is full or corrupt. From an AS/400 5250 session, enter:

```
WRKOBJ OBJ(*ALL) OBJTYPE(*SQLPKG)
```

Verify size of package. The limit is 16 MB on a V4R2M0 system. There is no limit on V4R3M0.

22.5.7 SQL packages beginning with an underscore or special characters

The system security level must be set to 30 or less. Enter the following command:

```
CALL QSYS/QLIDLOBJ ('package name library name' 'SQLPKG')
```

The package name and the library name are both blank-padded case-sensitive 10 character fields. There should be no delimiter specified between the names, and the whole 20 character string should be enclosed in single quotes. The second parameter is also enclosed in single quotes, for example:

```
CALL QSYS/QLIDLOBJ ('_CSETUP QGPL' 'SQLPKG')
```

22.5.8 Client Access Express diagnostics

This tip outlines the various traces, logs, and utilities that are part of or can be used with Client Access Express to help isolate problems that occur while running or within the Client Access Express product. They can also be used to help determine the cause of problems within an application using APIs in the Client Access Express product.

22.5.8.1 Client Access traces and logs

This section describes various Client Access problem diagnostic tools.

Diagnostic tools

The Diagnostic Tools utility provides a unified interface for starting, stopping, displaying, and working with the properties of the history log, detail trace, and entry point trace.

- **Usage:** Double-clicking the Start Diagnostic Tools icon in the Client Access Express Service folder launches an icon in the Windows system tray (normally displayed in the lower right corner of the screen). Select this icon to start or stop the diagnostic tools and change their properties.
- **Additional information:**
 - If the system has a great deal of activity, you will want to increase the size the logs from the default. This prevents situations where a large number of informational messages in the history log wrap and overlay an error message that you may want to investigate.
 - The history log is a single log for the entire system on a Terminal Server.
 - Separate logs are maintained for entry or exit and detailed traces if you are using Terminal Server, but only if each user changes the settings for the log file name to use in the setup for the trace.

History log

Log for messages that occur while Client Access Express is running on a machine. Similar to QSYSOPR message queue on the AS/400 system, this log contains information for all activity and is not separated out by the particular program. The messages may be informational, warning, or error messages. If an error occurs and no other traces are running, this log is a good place to look for possible additional information on what caused the error.

- **Usage:** Start this logging through the Client Access Diagnostic Tool. The tool can also be used to change its properties such as the size of the log and whether or not it will wrap. To view this log, in the Service Folder of Client Access Express, select the **History Log** icon. This opens the history log.
- **Additional information:** When a message in the history log includes a message ID, you can double-click it to display the secondary help text for the message. The secondary help text contains cause and recovery information for the message.

Entry/exit trace

This utility logs both external APIs and internal APIs as they enter a function and as they exit. Exits also log the return code that they exit with. If your application calls Client Access Express APIs, this function may be helpful determining where the application code is failing

- **Usage:** Start this trace through the Client Access Diagnostic Tool. The tool can also be used to change its properties such as the size and wrap as well as setting filters to only log information for particular Client Access Express components. To view, in the Service Folder of Client Access Express, select the **Traces** folder and then select **Entry/Exit Trace**. This opens the trace file after it has completed.
- **Additional information:**
 - Not all Client Access Express functions use this tracing facility. ODBC is one function that does not use this trace. ODBC tracing can be performed in this case.
 - This trace function *will* impact performance. Keep this in mind if you want to use this tool to debug errors that are timing related.
 - This trace function has been useful determining problems where a PC cannot make any type of connection to an AS/400 system.

Detail trace

This utility logs detailed information at specific key points in Client Access Express components' function. The type and amount of information vary between components. This trace is most useful for Client Access developers.

- **Usage:** Start this trace through the Client Access Diagnostic Tool. The tool can also be used to change its properties such as the size and wrap as well as setting filters to only log information for particular Client Access Express components. To view this log, in the Service Folder of Client Access Express, select the **Traces** folder and then select **Detail Trace**. This opens the trace file.
- **Additional information:**
 - Not all Client Access Express functions use this tracing facility. ODBC is one function that does not use this trace. ODBC tracing can be performed in this case.
 - This trace function *will* impact performance. Keep this in mind if you want to use this tool to debug errors that are timing related.
 - This trace function has been useful determining problems where a PC cannot make any type of connection to an AS/400 system.

CWBPING

This utility verifies connections between the PC and any AS/400 system. It will verify that the PC can connect to the various host server programs that Client Access Express uses.

- **Usage:** The utility is in the Windows directory of your PC (for example, c:\WINNT). Run the `CWBPING <as400-name>` command to verify connections with all host server programs. Other options are available to use specific signon information, SSL, and so on.
- **Additional information:** This utility is useful to verify that a connection can be established between the PC and the AS/400 server program. For example, ODBC connections can be established if the message `Successfully connected to server application - Data Access` is displayed as part of `cwbping` running.

Client Access communication trace (CWBCOTRC)

Sometimes more communications information than the Client Access Express Detail and Entry/Exit Trace gathers is needed to determine the cause of a problem. This trace is most useful for the Client Access developers and is not as useful for application debugging.

- **Usage:** This client communications trace is started and stopped by running `cwbcotrc.exe` from a DOS prompt while in the Client Access installation directory. Type `cwbcotrc` without any options to see the command format.
- **Additional information:** The trace file does *not wrap*. It just keeps growing. After you have run this trace as long as necessary, make sure you turn it off.

Sock Spy

This tool allows the tracing of most of the Winsock 1.1/2.x APIs, either on a system-wide or application-specific basis. It works simply by replacing the `WSOCK32.DLL` with a new version that intercepts calls to Winsock, logs entry and input data, calls the real `WSOCK32.DLL`, and logs output data, and exits on the way back.

- **Usage:** To use this tool, replace the wsock32.dll module in the winnt/system32 directory (note, “winnt” is the location that Windows NT is installed, your directory name may be different). You need to boot in DOS or use a tool like BOOTMOVE to replace this DLL. Complete usage instructions are contained in a text file that comes with the tool cwbwsspy.txt.
- **Additional information:**
 - The trace file does not wraparound. It continues to grow until available disk space is filled *or* until trace is turned off.
 - All sockets information will be traced, so this tool can be used to trace Client Access Express communications or your application’s communication through sockets.
 - This is an internal IBM development trace. To use it, you need to a copy from IBM.

22.5.8.2 Other traces, logs, and utilities

This section describes miscellaneous problem identification tools.

ODBC trace

The ODBC trace allows you to create logs of the calls to ODBC drivers to aid in debugging either the application or the ODBC driver.

- **Usage:** From the 32-bit ODBC Administrator select the **Tracing** tab. Click the **Start Tracing** button to begin the trace. Reproduce the scenario that you want traced, and click the **Stop Tracing** button. By default, the log is located in the root of the drive.
- **Additional information:**
 - This trace is useful for ODBC driver providers, such as Client Access Express, to follow the flow of SQL calls that are being issued on behalf of the client application.
 - This trace is also useful for application providers to determine the calls that they are making to the ODBC driver and the resulting SQL calls.

Bootmove

Some modules are used by applications as soon as the system boots. If new versions or debug versions need to be installed on a system, these modules must be replaced before they are locked by using applications. An example is the Client Access Express communications module cwbcows.dll. Bootmove is a tool that replaces files while booting a machine, so they are in place before they are in use. The alternative is to boot in DOS or find a way to end all applications that are using the module.

- **Usage:** BOOTMOVE <new module> <existing module>.
- **Note:** You need to enter the complete path for both the new and existing module.
- **Additional information:** Bootmove moves the module, meaning that it replaces the existing module. You should make a copy of the existing module before running bootmove.

22.5.9 ODBC dropped connections

If an ODBC connection is dropped, perform this process:

1. Search for job logs in the spooled file and output queue QEZJOBLOG. Look for messages in QSYSOPR and the user message queue.
2. Check for VLOG entries.
3. Start a Communications Trace.
4. Start an ODBC Trace.

22.6 Remote support preparation

All problems and defect reports are usually channelled through the Global Support organization. The normal process of defect management and escalation is followed by IBM and J.D. Edwards. You should report the problem to the appropriate support structure when a problem solution is needed.

Refer to Appendix E, "Support structure" on page 823, for the defect support structure for IBM AS/400 and J.D. Edwards Customer Support.

22.6.1 Preparing for IBM Remote Support

You need to complete the following steps as a prerequisite to AS/400 system Remote Support. This allows access from IBM through the ECS link. The steps are:

1. Ensure the system is at a current PTF CUM (cumulative) level and any required PTFs are installed.
2. For IBM to sign on to the system, you must create a user profile on the AS/400 system with QSECOFR authority. If IBM has access to OneWorld through J.D. Edwards OneWorld, you also need a User ID in OneWorld with super user authority.
3. The user profile that IBM uses to sign on to the system must be enrolled in the system distribution directory for IBM to use the SNADS functions (or the ability to send data to and from the customer's system). Check this by using the AS/400 system Work with Directory Entries (`WRKDIR`) command. Select option 1, Add, to add a user if they are not already in the system distribution directory.
4. If IBM wants to use SNADS, they need to know the current system name. This can be checked using the AS/400 system Display Network Attributes (`DSNETA`) command.
5. Determine which AS/400 system hardware resource that the ECS line QESLINE is using. This is usually LIN011, but may be different, such as CMN11, depending on the customer's configuration. Check this by using the AS/400 system Display Line Description (`DSPLIND`) command:

```
DSPLIND LIND(QESLINE)
```
6. The hardware resource mentioned in the previous step must not be in use by anyone else and must be varied off before IBM can use it. This can be checked using the AS/400 system Work with Hardware Resources (`WRKHDWRSC`) command and specifying that you want to look at the communications hardware. Enter:

WRKHDWRSC TYPE(*CMN)

Enter option 5, Work with configuration descriptions, next to the resource, and enter option 8, Work with configuration status, next to each configuration description. Use option 2, Vary off, to vary off anything that is varied on.

7. The phone number for the ECS modem should be determined and written on a location such as on the modem or the phone jack that the modem plugs into since this information is not located anywhere on the AS/400 system. This is the number IBM dials to enter into the system. The modem should also be properly configured. For example, if you have an IBM 7855 modem, it should be in synchronous mode.
8. The system IBM wants to access may not have a modem attached for some reason or IBM may have problems connecting. As an alternative solution, have IBM dial into a different AS/400 system (if available) and Telnet or pass through to the desired system. This is not the desired connection by IBM.
9. After everything is set up correctly and the information IBM needs to sign on is ready, use the AS/400 system Start Remote Support (`STRRMTSPT`) command. The values for all parameters are given to the customer by IBM since they change each time IBM makes a connection. The End Remote Support (`ENDRMTSPT`) command is used to clean everything up.

22.6.2 Preparing for J.D. Edwards remote support

To most effectively support your client/server solutions, J.D. Edwards Customer Support consultants may need access to your client desktops, Deployment Servers, and Enterprise Servers. The consultants gain access to your environment by remote dial-up. You control and secure the access to your environment by having control over the communication links used to dial in and the user profiles used to obtain access. The Customer Support consultant may use the J.D. Edwards developed facility, JDELINK, to access your AS/400 server. Or, if it is necessary to access your Deployment Server or a client workstation, the consultant uses ControllIT, a third-party remote connectivity product provided by Computer Associates (<http://www.cai.com/products/controlit.htm>).

22.6.2.1 JDELINK

JDELINK is a set of communications software to facilitate AS/400 communications from J.D. Edwards. The software is self-contained in a library named JDELINK2. It does not depend on any release level of J.D. Edwards software. However, your IBM operating system (OS/400) must be at a version level currently recommended by J.D. Edwards. JDELINK lets J.D. Edwards dial into your AS/400 system to diagnose and solve problems.

JDELINK is shipped with J.D. Edwards WorldSoftware. If you are a OneWorld only user, you can contact J.D. Edwards Technical Customer Support to be sent a savefile containing JDELINK.

Part 4. Appendixes

This part contains appendixes and complementary information to the chapters. This part includes these appendixes:

- Appendix A, “Adding a user auxiliary storage pool” on page 775, shows how to create a user Auxiliary Storage Pool (user ASP), which is a highly recommended place for you to place all journal receivers.
- Appendix B, “TCP/IP basic installation and configuration” on page 781, describes the process of configuring Transmission Control Protocol/Internet Protocol (TCP/IP) on the AS/400 system.
- Appendix C, “National language versions on AS/400 systems” on page 817, provides a brief overview of some useful information related to national language versions on AS/400 system, such as CCSID, character ID, keyboard ID, and so on.
- Appendix D, “Interprocess communication (IPC)” on page 821, describes the OS/400 services that provide the OneWorld processes a way to communicate with each other through a set of standardized APIs.
- Appendix E, “Support structure” on page 823, outlines the support available for OneWorld on the AS/400 system and also contains contact information for customers and business partners.
- Appendix F, “Complementary products” on page 843, contains a list of J.D. Edwards' business partners, which offer add-on products for OneWorld on the AS/400 system.

The remainder of this part contains such useful reference information as the bibliography, glossary of terms, and index.

Appendix A. Adding a user auxiliary storage pool

Creating a user auxiliary storage pool (user ASP) allows the available disk space to be partitioned and managed as separate entities. For example, you may allocate one disk unit to a particular ASP and thus always know on which disk unit your data resides. In an AS/400 system environment, we strongly recommend that a user ASP is created to house the journal receivers produced during database activity.

There are two methods of creating a user ASP. The first method is to create the ASP when new disks are added to the system. These disks can be configured directly into the new ASP. The second method is to move existing and populated disks from the existing system or user ASPs into the new ASP. This is a more lengthy process because moving populated disks requires that the resident data be copied onto the remaining disks of the originating ASP.

Note: The creation of a user ASP can *only* be performed through Dedicated Service Tools (DST), which must be accessed during a manual or attended IPL of the AS/400 system.

This appendix details the steps involved in the creation of a user ASP when moving existing unprotected disks into a new ASP. There are additional considerations when moving populated disks on systems where disk protection is in place.

Note

This appendix shows how to create ASP in OS/400 V4R4. Operations Navigator in OS/400 V4R5 brings the graphical user interface that you can use to perform the same tasks.

For more information on ASPs or any of the possible ASP creation scenarios, refer to the manual entitled *Backup and Recovery*, SC41-5304.

A.1 Checking the disk configuration

An easy way to determine the current disk configuration is to use the AS/400 system Work with Disk Status (WRKDSKSTS) command. Figure 423 on page 776 shows an example of the information you may see.

```

Work With Disk Status

Elapsed time: 00:00:00

Unit  Type  Size  %  I/O  Request  Read  Write  Read  Write  %
      (M)  Used  Rqs  Size (K)  Rqs  Rqs  (K)  (K)  Busy
  1  6606  1967  98.4  .0  .0  .0  .0  .0  .0  0
  1  6606  1967  98.4  .0  .0  .0  .0  .0  .0  0
  2  6713  7516  58.6  .0  .0  .0  .0  .0  .0  0
 11  6713  7516  58.1  .0  .0  .0  .0  .0  .0  0
 12  6713  7516  58.0  .0  .0  .0  .0  .0  .0  0
 13  6713  7516  58.1  .0  .0  .0  .0  .0  .0  0
 14  6713  7516  58.1  .0  .0  .0  .0  .0  .0  0
 15  6713  7516  58.0  .0  .0  .0  .0  .0  .0  0
 16  6713  7516  58.0  .0  .0  .0  .0  .0  .0  0
 17  6713  7516  58.0  .0  .0  .0  .0  .0  .0  0

Bottom
Command
====>
F3=Exit  F5=Refresh  F12=Cancel  F24=More keys

```

Figure 423. Work with Disk Status (WRKDSKSTS) display

If you press the F11 key once, you can see additional information such as in which configured ASPs the disk units are located and what disk protection (if any) they are using. Figure 424 shows an example of the information you may see.

```

Work with Disk Status

Elapsed time: 00:00:00

--Protection--
Unit  ASP  Type  Status  Compression
  1    1  MRR  ACTIVE
  1    1  MRR  ACTIVE
  2    1  DPY  ACTIVE
 11    1  DPY  ACTIVE
 12    1  DPY  ACTIVE
 13    1  DPY  ACTIVE
 14    1  DPY  ACTIVE
 15    1  DPY  ACTIVE
 16    1  DPY  ACTIVE
 17    1  DPY  ACTIVE

Bottom

Command
====>
F3=Exit  F5=Refresh  F12=Cancel  F24=More keys

```

Figure 424. Work with Disk Status (WRKDSKSTS) display: ASP and disk protection

For example, the display in Figure 424 indicates that all the disks are in ASP 1 (the system ASP). All except the first two, which are mirrored (MRR), are protected using RAID-5 or device parity protection (DPY).

A.2 Configuring a new user ASP

Reallocation of disk drives to create a new user ASP requires a dedicated system. The time required to perform this operation can take from a few minutes to a few hours depending on the number of disks installed and the amount of data on the system.

Note

Ensure that you have the Dedicated Service Tools (DST) user identification and password. Contact your computer system's security officer or your IBM Engineer for this information.

A.2.1 Starting Dedicated Service Tools

If the AS/400 system is powered on and active, you need to ensure that all users have signed off from the system. Then, perform the following steps:

1. Place the AS/400 key or keystick in the control panel if applicable.
2. Place the system in Manual mode on the control panel.
3. Power down the system with the AS/400 system Power Down System (PWRDWNSYS) command with *restart* set to *YES. Figure 425 shows an example of the PWRDWNSYS command.

```
Power Down System (PWRDWNSYS)

Type choices, press Enter.

How to end . . . . . *IMMED          *CNTRLD, *IMMED
Delay time, if *CNTRLD . . . . . 3600          Seconds, *NOLIMIT
Restart options:
  Restart after power down . . . *YES          *NO, *YES
  Restart type . . . . .          *IPLA, *SYS, *FULL
IPL source . . . . . *PANEL          *PANEL, A, B, D

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
```

Figure 425. Power Down System (PWRDWNSYS) display

4. When the display shown in Figure 426 appears during the manual IPL, select option **3** (Use Dedicated Service Tools (DST)).

```
IPL or Install the System

Select one of the following:
1. Perform an IPL
2. Install the operating system
3. Use Dedicated Service Tools (DST)
4. Perform automatic installation of the operating system
5. Save Licensed Internal Code
```

Figure 426. Manual IPL: IPL or Install the System display

- The display in Figure 427 appears. Sign on using the appropriate Dedicated Service Tools (DST) user profile and password.

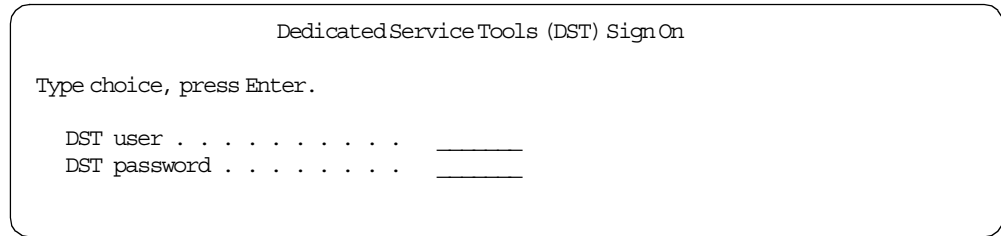


Figure 427. Dedicated Service Tools (DST) Sign On display

- When the Use Dedicated Service Tools (DST) menu appears, as shown in Figure 428, select option **4 (Work with disk units)**.

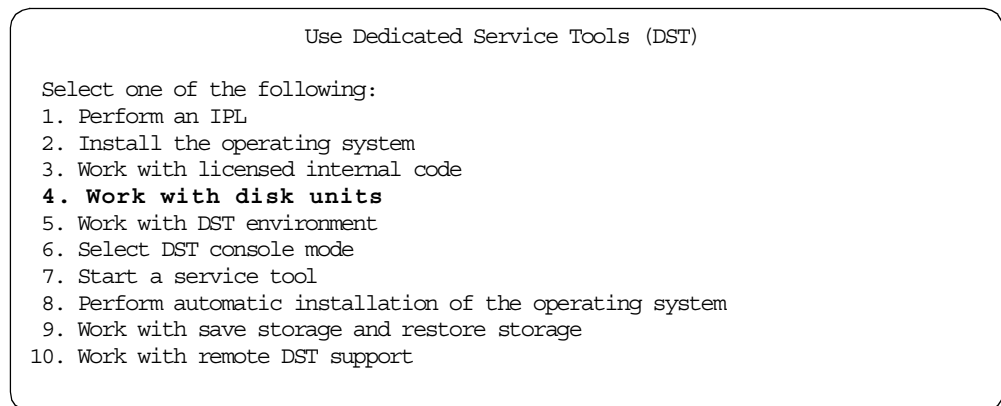


Figure 428. Dedicated Service Tools (DST) main menu display

- From the next menu, select option **1 (Work with disk configuration)**, option **3 (Work with ASP configuration)**, and option **6 (Move units from one ASP to another)**. A menu (Figure 429) appears that allows you to select the disk drives to be included in the new user ASP.

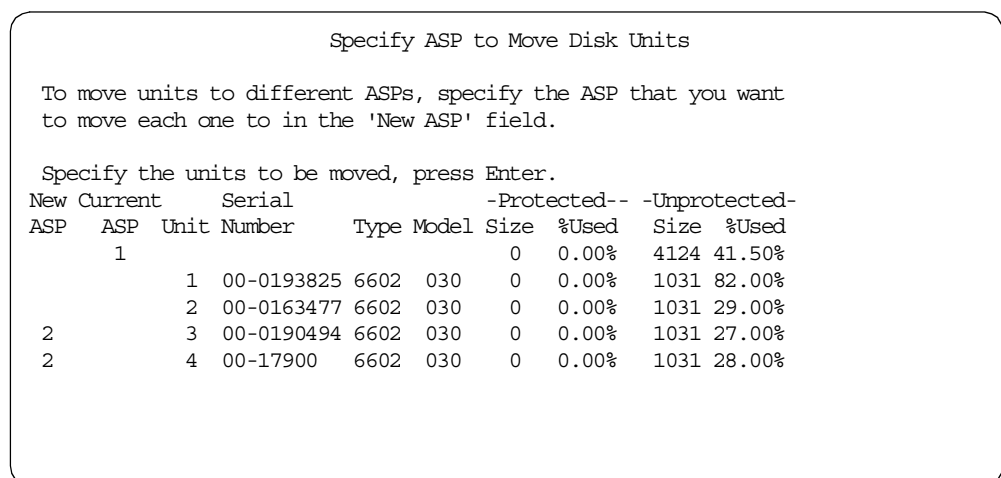


Figure 429. DST Specify ASP to Move Disk Units display

Note

If you are moving multiple disk drives to a new ASP, try to choose disk drives that are spread over multiple disk IOPs and in multiple RAID-5 sets. This offers better system performance.

8. Confirm your choices in the resulting displays. The system sets up the new user ASP. All disk drives are moved in parallel. This may take some time to complete.
9. When the process is complete, you can press the F3 or F12 keys to return to the main DST menu shown in Figure 426 on page 777. Before starting the IPL, we recommend that you put the AS/400 system into Normal mode on the control panel. After the IPL completes, the user ASP is ready for use.

Appendix B. TCP/IP basic installation and configuration

This appendix describes the process of configuring Transmission Control Protocol/Internet Protocol (TCP/IP) on the AS/400 system. If you want extensive knowledge about TCP/IP on the AS/400 platform, refer to *V4 TCP/IP for AS/400: More Cool Things Than Ever*, SG24-5190, *OS/400 TCP/IP Configuration and Reference*, SC41-5420, and *TCP/IP Fastpath Setup*, SC41-5430.

Note

You should configure TCP/IP prior to installing the J.D. Edwards OneWorld.

B.1 Configuring TCP/IP on the AS/400 system

You can configure and work with TCP/IP using Operations Navigator or the Command Line interface. For some functions, the entire configuration must be done through Operations Navigator, while other functions can only be configured through the command line interface.

B.1.1 TCP/IP configuration using a command line interface

This section shows you how to configure a basic TCP/IP connection using an AS/400 command line interface. You need one local workstation connected to the AS/400 system to do this.

The configuration process is complete once you perform a PING successfully on the AS/400 system. If you need more information, see *TCP/IP Fastpath Setup*, SC41-5430.

Here is a simple Ethernet network to show you how to configure TCP/IP. In Figure 430, you see an Ethernet network with one AS/400 system and one personal computer. The configuration is shown from the AS/400 system perspective.

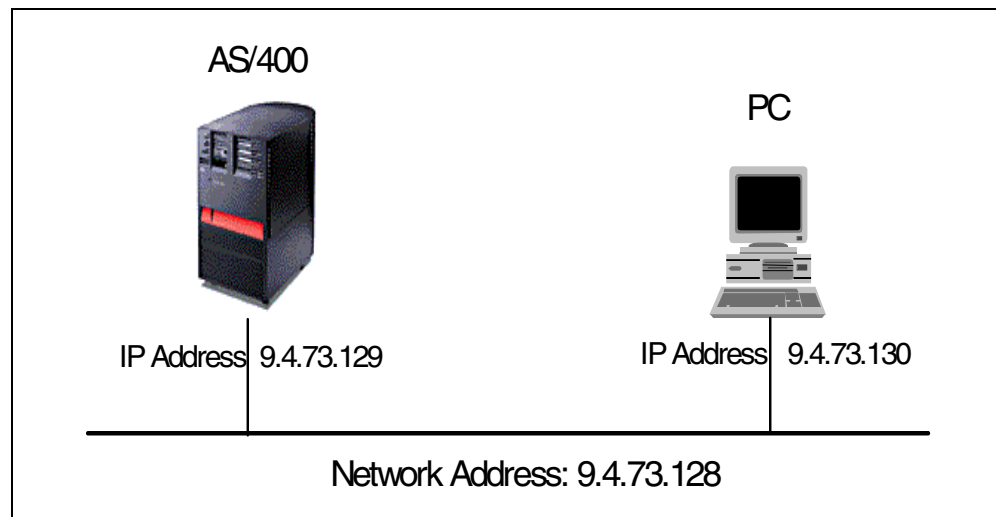


Figure 430. Ethernet network example

B.1.1.1 Collecting the network parameters

You need to know the addresses and subnet mask of your AS/400 system, router, and gateway. You also need your local domain name and the host name of your AS/400 system. In this scenario, we used the following information:

- **Network address:** 9.4.73.128
- **Subnet mask:** 255.255.255.128
- **Domain name:** SYSNAM123.IBM.COM
- **AS/400 host Internet address/name:** 9.4.73.129/SYSNAMA

Note

If you are the network administrator and if you never plan to attach your network to the Internet, we suggest that you use an address such as 128.1 for your TCP/IP network and 128.1.0.1 for your first TCP/IP host. The subnet mask would be 255.255.0.0.

B.1.1.2 Signing on to the AS/400 system

Sign on the AS/400 system as user QSECOFR (or another user with the special authority of *IOSYSCFG).

Note

System configuration (*IOSYSCFG) special authority gives the user the ability to change how the system is configured. This may include, for example, adding or removing communications configuration information, working with TCP/IP servers, and configuring the Internet Connection Server (ICS).

B.1.1.3 Creating a line description

In this scenario, an Ethernet line is used. To see other types of lines, refer to Appendix A, "Configuring a Physical Line for TCP/IP communications" in *OS/400 TCP/IP Configuration and Reference*, SC41-5420.

You need to know the resource name and decide the line name. To do this, perform the following steps:

1. The resource name is the AS/400 system name for the adapter card. To obtain the resource name, enter the Work with Hardware Resources (WRKHDWRSC) command and specify the communications resources as shown here:

```
WRKHDWRSC TYPE(*CMN)
```

When you press Enter, the display shown in Figure 431 appears.

Work with Communication Resources				
Type options, press Enter.				
5=Work with configuration descriptions			7=Display resource detail	
Opt	Resource	Type	Status	Text
	CC13	2629	Operational	Comm Processor
	LIN20	6181	Operational	LAN Adapter
	CMN24	6181	Operational	Ethernet Port
	LIN22	6149	Operational	LAN Adapter
	CMN25	6149	Operational	Token-Ring Port
	LIN28	2699	Operational	Comm Adapter
	CMN29	2699	Operational	V.24 Port
	CMN30	2699	Operational	Comm Port
	CC14	6617	Operational	File Server IOP
	LIN30	2723	Operational	LAN Adapter
	CMN32	2723	Operational	Ethernet Port
	LIN33	2724	Operational	LAN Adapter
	CMN33	2724	Operational	Token-Ring Port
				More...
F3=Exit F5=Refresh F6=Print F12=Cancel				

Figure 431. Work with Communication Resources (WRKHDWRSC TYPE(*CMN))

2. Choose the proper resource name from the display, and decide the name of the line for the line description.
3. To create a line description, enter the Create Line Description (Ethernet) (CRTLINETH) command.

Fill in the line name and resource name, and press Enter. The display shown in Figure 432 on page 784 appears. In this example, the resource name is CMN24, and the line name is LINETH01.

Note: Step 2 and step 3 can also be done in the following way:

Type option 5 (Work with configuration descriptions) in the Opt field as shown in Figure 432 on page 784. Then, the Work with Configuration Descriptions display appears. In this display, type option 1 (Create) in the Opt field. The display shown in Figure 432 on page 784 appears.

```

                                Create Line Desc (Ethernet) (CRTLINETH)

Type choices, press Enter.

Line description . . . . . > LINETH01      Name
Resource name . . . . . > CMN24           Name, *NWID, *NWSID
Online at IPL . . . . . *YES              *YES, *NO
Vary on wait . . . . . *NOWAIT           *NOWAIT, 15-180 seconds
Local adapter address . . . . . *ADPT     020000000000-7FFFFFFF...
Exchange identifier . . . . . *SYSGEN     05600000-056FFFFF, *SYSGEN
Ethernet standard . . . . . *ALL          *ETHV2, *IEEE8023, *ALL
Line speed . . . . . 10M              Character value, 10M, 100M...
Duplex . . . . . *FULL                Character value, *HALF...
SSAP list:
  Source service access point . *SYSGEN   02-FE, *SYSGEN
  SSAP maximum frame . . . . .           *MAXFRAME, 265-1496, 265...
  SSAP type . . . . .                   *CALC, *NONSNA, *SNA, *HPR
    + for more values
Text 'description' . . . . . *BLANK

                                                                More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 432. Create Line Description (Ethernet) (CRTLINETH) display

Ensure that the modes and line speed for the Ethernet line are matched correctly, for example, if a switch is set to full duplex, and the AS/400 system is set to full duplex.

B.1.1.4 Creating a TCP/IP interface

You need the IP and subnet mask addresses to create a TCP/IP interface. Perform the following steps:

1. Go to the CFGTCP menu, and select option 1 (Work with TCP/IP Interfaces). The display shown in Figure 433 appears.

```

                                Work with TCP/IP Interfaces

Type options, press Enter.
  1=Add  2=Change  4=Remove  5=Display  9=Start  10=End

      Internet      Subnet      Line      Line
Opt  Address      Mask      Description  Type

      127.0.0.1    255.0.0.0  *LOOPBACK  *NONE

                                                                More...
F3=Exit  F5=Refresh  F6=Print list  F10=Work with IP over SNA interfaces
F11=Display interface status  F12=Cancel  F17=Top  F18=Bottom

```

Figure 433. CFGTCP menu display: Work with TCP/IP Interfaces

2. Select option 1 (Add) to add a TCP/IP interface to the table, and press Enter. The display shown in Figure 434 appears.

```

                                Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . > '9.4.73.129'
Line description . . . . . lineth01      Name, *LOOPBACK
Subnet mask . . . . . 255.255.255.128
Associated local interface . . . *NONE
Type of service . . . . . *NORMAL      *MINDELAY, *MAXTHRPUT...
Maximum transmission unit . . . *LIND      576-16388, *LIND
Autostart . . . . . *YES      *YES, *NO
PVC logical channel identifier . . . . . 001-FFF
      + for more values
X.25 idle circuit timeout . . . 60      1-600
X.25 maximum virtual circuits . 64      0-64
X.25 DDN interface . . . . . *NO      *YES, *NO
TRLAN bit sequencing . . . . . *MSB      *MSB, *LSB

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 434. Add TCP/IP Interface (ADDTCPIFC) display

3. Enter the Internet address, line description, and subnet mask. After pressing Enter, the Work with TCP/IP Interface display appears as shown in Figure 435. Note the message at the bottom that states the TCP/IP interface was added successfully.

```

                                Work with TCP/IP Interfaces

Type options, press Enter
  1=Add  2=Change  4=Remove  5=Display  9=Start  10=End

Opt  Internet      Subnet      Line      Line
     Address      Mask        Description Type

     9.47.73.129   255.255.255.128  LINETH01  *ELAN
     127.0.0.1     255.0.0.0      *LOOPBACK *NONE

More...
F3=Exit  F5=Refresh  F6=Print list  F10=Work with IP over SNA interfaces
F11=Display interface status  F12=Cancel  F17=Top  F18=Bottom
TCP/IP interface added successfully.

```

Figure 435. CFGTCP menu display: Work with TCP/IP Interfaces

B.1.1.5 Creating a TCP/IP host table entry

A host table entry allows you to associate a host name (for example, SYSNAMA) to an Internet address (for example, 9.47.73.129). To do this, perform the following tasks:

1. Select option **10** (Work with TCP/IP host table entries) on the CFGTCP menu, and press Enter. The display shown in Figure 436 on page 786 appears.

```

Work with TCP/IP Host Table Entries

Type options, press Enter.
  1=Add  2=Change  4=Remove  5=Display  7=Rename

      Internet      Host
Opt  Address       Name

      127.0.0.1     LOOPBACK

Bottom

F3=Exit  F5=Refresh  F6=Print list  F12=Cancel  F17=Position to

```

Figure 436. CFGTCP menu display: Work with TCP/IP Host Table Entries

2. Select option 1 (Add) to add one entry to the table. The Add a TCP/IP Host table entry display should appear. This is shown in Figure 437.

```

Add TCP/IP Host Table Entry (ADDTCPHTE)

Type choices, press Enter.

Internet address . . . . . > 9.4.7.129
Host names:
  Name . . . . . > SYSNAMA

  Name . . . . . > SYSNAMA.SYSNAM123.IBM.COM
Text 'description' . . . . . 'Entry for AS/400 (SYSNAMA) '

Bottom

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 437. Add TCP/IP Host Table Entry (ADDTCPHTE) display

3. This display provides fields for the Internet address, associated host name, and an optional text description. Type in the values. After pressing Enter, the Work with TCP/IP Host Table Entries appears as shown in Figure 438.

```

Work with TCP/IP Host Table Entries

Type options, press Enter.
  1=Add  2=Change  4=Remove  5=Display  7=Rename

      Internet      Host
Opt  Address       Name

      9.4.73.129    SYSNAMA
      127.0.0.1     LOOPBACK
      127.0.0.1     SYSNAMA.SYSNAM123.IBM.COM

Bottom

F3=Exit  F5=Refresh  F6=Print list  F12=Cancel  F17=Position to

```

Figure 438. CFGTCP menu display: Work with TCP/IP Host Table Entries

For information on how to configure a TCP/IP route, refer to B.2.2, “Route configuration” on page 815.

B.1.1.6 Configuring a local domain and host name

The local host and domain name are used to identify the AS/400 system within the TCP/IP domain in which it is located. POP and SMTP mail servers require that the local host and domain name be configured.

To define the local host and domain name for AS/400 machine, select option **12** (Change TCP/IP domain information) from the CFGTCP menu. The display shown in Figure 439 appears.

```
Change TCP/IP Domain (CHGTCPDMN)

Type choices, press Enter.

Host name . . . . . 'SYSNAMA'

Domain name . . . . . 'SYSNAM123.IBM.COM'

Host name search priority . . . *LOCAL      *REMOTE, *LOCAL, *SAME

Internet address . . . . . '9.4.73.129'

Additional Parameters

Port . . . . . 53          1-65535, *SAME
Protocol . . . . . *UDP      *UDP, *TCP, *SAME

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
```

Figure 439. Change TCP/IP Domain (CHGTCPDMN) display

In this example, 9.4.73.129 is the IP address of the DNS server.

For more information about the local host name and domain name, see *AS/400 TCP/IP Configuration and Reference*, SC41-5420.

B.1.1.7 Starting TCP/IP

Before any TCP/IP services are available, TCP/IP processing must be initialized and activated. The steps for starting TCP/IP are:

1. Select option **3** (Start TCP/IP) from the TCP/IP Administration menu using the command:

```
GO TCPADM
```

You can also choose to enter the Start TCP (`STRTCP`) command. Both commands initialize and activate TCP/IP processing. They also start the TCP/IP interface control job, which starts all interfaces with an AUTOSTART value set to `*YES`. It starts the TCP/IP server jobs.

2. Allow a few moments for TCP/IP to start. Then, enter the command:

```
WRKACTJOB SBS(QSYSWRK) JOB(QT*)
```

The job QTCPIP should appear as shown in Figure 440 on page 788. After this job is started, you can proceed with TCP/IP connection verification.

```

Work with Active Jobs

CPU %:    2.2    Elapsed time:  00:01:21    Active jobs:  219

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files  13=Disconnect ...

Opt  Subsystem/Job  User      Type  CPU %  Function      Status

      QTCPIP       QTCP      BCH    .0      DEQW
      QFTFP01357   QTCP      BCH    .0      CMD-CHGJOB    ICFW
      QTGTELNETS   QTCP      BCH    .0      DEQA
      QTLPD00266   QTCP      BCH    .0      DEQW
      QTMSNMP      QTCP      BCH    .0      PGM-QTOSMAIN  DEQW
      QTSMTPSRVR   QTCP      BCH    .0      PGM-QIMSSRCP  SELW
      QTWSG02808   QTIMWSG   BCH    .0      TIMW

Bottom

Parameters or command
====>
F3=Exit   F4=Prompt   F5=Refresh  F10=Restart statistics
F11=Display elapsed data  F12=Cancel  F14=Include  F24=More keys

```

Figure 440. Work with Active Jobs (WRKACTJOB) display - QTCPIP job

Messages indicating that TCP/IP has been started are also sent to the QTCP and QSYSOPR message queues. To check for the successful start of TCP/IP, enter either of these commands:

```

DSPMSG QSYSOPR
DSPMSG QTCP

```

If the QTCPIP job does not start, look for spooled job logs. Generally, the user for these job logs is QTCP. Use the Work with Spooled Files (WRKSPLF) command, and specify QTCP for the user to find the logs:

```

WRKSPLF QTCP

```

B.1.1.8 Stopping TCP/IP

Stopping TCP/IP ends all TCP/IP processing, all active TCP/IP interfaces, and all TCP/IP connections on the AS/400 system with which you are working. Unless you specified `ENDSVR (*NO)`, all TCP/IP server jobs for agents that are currently active in the QSYSWRK subsystem are ended. There are two possible values when stopping TCP/IP: *controlled* and *immediately*. There are two ways to stop TCP/IP using the CL interface:

- Enter `ENDTCP`, and press F4.
- Type `GO TCPADM`, and select option 4 (End TCP/IP).

To stop TCP/IP from the TCP/IP Administration menu, perform the following steps:

1. Type `GO TCPADM` from the AS/400 Main Menu. The TCP/IP Administration Menu is displayed (Figure 441).


```

TCPADM                                TCP/IP Administration

Select one of the following:

    1. Configure TCP/IP
    2. Configure TCP/IP applications
    3. Start TCP/IP
    4. End TCP/IP
    5. Start TCP/IP servers
    6. End TCP/IP servers
    7. Work with TCP/IP network status
    8. Verify TCP/IP connection
    9. Start TCP/IP FTP session
   10. Start TCP/IP TELNET session
   11. Send TCP/IP spooled file

    20. Work with TCP/IP jobs in QSYSWRK subsystem

Selection or command
====> 4

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 1998.

```

Figure 441. TCP/IP Administration menu

2. Select option 4 (End TCP/IP), and press Enter. The End TCP/IP display is shown (Figure 442).

```

                                End TCP/IP (ENDTCP)

Type choices, press Enter.

How to end . . . . . *IMMED      *IMMED, *CNTRL D

                                Additional Parameters

End application servers . . . . *YES      *YES, *NO

                                Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 442. End TCP/IP (ENDTCP)

Note: There is no confirmation display shown when ENDTCP is entered. The ENDTCP command must be used with caution. When it is used, it ends all TCP/IP processing on the AS/400 system on which you are working.

3. To individually stop a TCP/IP server, use the ENDTCPSVR command, and press F4. The End TCP/IP Server display is shown in Figure 443 on page 790.

```

                                End TCP/IP Server (ENDTCPSVR)

Type choices, press Enter.

Server application . . . . . *ALL          *ALL, *SNMP, *ROUTED...
                             + for more values
                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 443. ENDTCPSVR

B.1.1.9 Verifying the TCP/IP connection

Use the PING command to verify the TCP/IP configuration. Perform the following steps:

1. To test the TCP/IP code without sending anything out of the Ethernet adapter, specify the special host name LOOPBACK:

```
PING LOOPBACK
```

The display shown in Figure 444 should appear.

```

                                Command Entry
                                Request level: 7

Previous commands and messages:
> ping loopback
  Verifying connection to host system LOOPBACK at address 127.0.0.1.
  PING request 1 from 127.0.0.1 took 57 ms. 256 bytes. TTL 64.
  PING request 2 from 127.0.0.1 took 0 ms. 256 bytes. TTL 64.
  PING request 3 from 127.0.0.1 took 0 ms. 256 bytes. TTL 64.
  PING request 4 from 127.0.0.1 took 0 ms. 256 bytes. TTL 64.
  PING request 5 from 127.0.0.1 took 0 ms. 256 bytes. TTL 64.
  Round-trip (in milliseconds) min/avg/max = 0/11/57
  Connection verification statistics: 5 of 5 successful (100 %).

                                                                    Bottom

Type command, press Enter.
====>

F3=Exit  F4=Prompt  F9=Retrieve  F10=Include detailed messages
F11=Display full  F12=Cancel  F13=Information Assistant  F24=More keys

```

Figure 444. PING LOOPBACK display: Successful Ping messages

2. To test the TCP/IP code, the Ethernet adapter, and the Ethernet LAN, specify the Internet address of the local adapter as defined in the host table:

```
PING RMTSYS(*INTNETADR) INTNETADR('9.4.73.129')
```

Or, enter:

```
PING RMTSYS(SYSNAMA)
```

TCP/IP sends data to the Ethernet adapter, and instructs the adapter to send the data to its own Ethernet address. If you have the correct configuration, the display shown in Figure 445 appears.

```

                                Command Entry
Request level:  4
Previous commands and messages:
> ping asm02
Verifying connection to host system RCHASM02.rchland.ibm.com at address
9.5.69.211.
PING request 1 from 9.5.69.211 took 34 ms. 256 bytes. TTL 64.
PING request 2 from 9.5.69.211 took 27 ms. 256 bytes. TTL 64.
PING request 3 from 9.5.69.211 took 9 ms. 256 bytes. TTL 64.
PING request 4 from 9.5.69.211 took 9 ms. 256 bytes. TTL 64.
PING request 5 from 9.5.69.211 took 10 ms. 256 bytes. TTL 64.
Round-trip (in milliseconds) min/avg/max = 9/17/34
Connection verification statistics: 5 of 5 successful (100 %).

                                                                Bottom

Type command, press Enter.
====>

F3=Exit   F4=Prompt   F9=Retrieve   F10=Include detailed messages
F11=Display full   F12=Cancel   F13=Information Assistant   F24=More keys

```

Figure 445. PING RMTSYS display: Successful Ping messages

If the PING execution ends unsuccessfully, the display shown in Figure 446 appears.

```

                                Command Entry
Request level:  5
Previous commands and messages:
> PING '194.140.3.1'
Verifying connection to host system 194.140.3.1.
No response from host within 1 seconds for connection verification 1.
No response from host within 1 seconds for connection verification 2.
No response from host within 1 seconds for connection verification 3.
No response from host within 1 seconds for connection verification 4.
No response from host within 1 seconds for connection verification 5.
Connection verification statistics: 0 of 5 successful (0 %).

                                                                Bottom

Type command, press Enter.
====>

F3=Exit   F4=Prompt   F9=Retrieve   F10=Include detailed messages
F11=Display full   F12=Cancel   F13=Information Assistant   F24=More keys

```

Figure 446. PING RMTSYS: Unsuccessful Ping messages

If you received unsuccessful PING messages, you should attempt the following tasks:

1. Check your configuration steps on the local system.
2. Check the configuration at the remote system.
3. Make sure the remote system is not powered down or TCP/IP is up and running.

If you have other error messages, see Appendix E, “TCP/IP Problem Analysis” in *OS/400 TCP/IP Configuration and Reference*, SC41-5420.

B.1.2 TCP/IP configuration using Operations Navigator

This section covers how to perform the following tasks for TCP/IP using Operations Navigator:

- Accessing a basic TCP/IP configuration
- Configuring a TCP/IP interface
- Configuring the domain and host name for TCP/IP
- Configuring host table entries for TCP/IP
- Configuring a TCP/IP route
- Starting and stopping TCP/IP
- Verifying a TCP/IP connection (Ping)

B.1.2.1 Accessing the TCP/IP configuration

AS/400 Operations Navigator is a powerful graphical interface for Windows 95, Windows 98, Windows NT, and Windows 2000 clients. To use Operations Navigator, you must have Client Access installed on your Windows 95, Windows 98, and Windows NT PC and have a connection to the AS/400 system that you want to configure. Refer to 11.2, "Operations Navigator and Management Central" on page 286, for more information on Operations Navigator.

TCP/IP allows you to connect an AS/400 system to a network. To reach the point where you can configure TCP/IP for your AS/400 system using Operations Navigator, perform the following steps:

1. Start Operations Navigator by clicking **Start->Programs->IBM AS400 Client Access->AS/400 Operations Navigator**. The AS/400 Operations Navigator window appears (Figure 447).
2. Double-click the AS/400 Systems icon (A). It gives you a list of all the AS/400 systems that you can access.
3. Double-click the AS/400 system (AS1) (B) that you want to configure.
4. Double-click **Networks** (C).
5. Double-click **Protocols** (D).
6. Right-click **TCP/IP** (E) in the right panel to see the context menu (Figure 447).

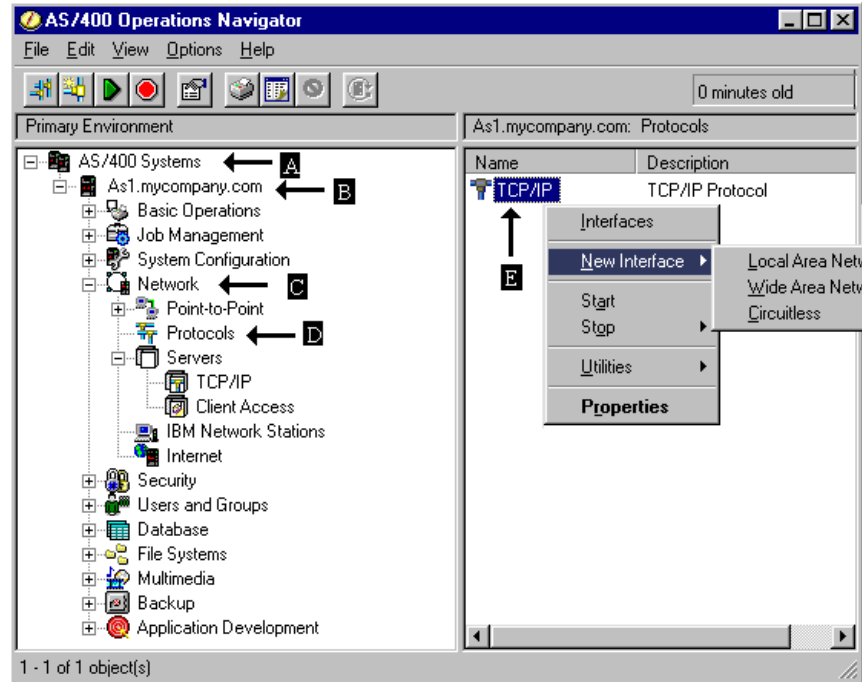


Figure 447. Operations Navigator protocols

You are now ready to start your configuration process.

B.1.2.2 Configuring a line for TCP/IP

The communication objects for AS/400 TCP/IP are the line descriptions, controller descriptions, and device descriptions. Operations Navigator allows you to configure a line for an Ethernet or a Token-Ring network adapter. When TCP/IP starts the line, controller and device descriptions are automatically varied on. If the controller and device descriptions do not exist, TCP/IP automatically creates them.

The procedure to create a line and to add TCP/IP support to it and the procedure to add TCP/IP support to an existing line are similar. In this section, we show a combination of both procedures with notes to point out where the differences occur.

The configuration wizard takes you through the steps that are needed to configure a line for TCP/IP for the AS/400 system. To use the configuration wizard, perform the following steps:

1. Access the TCP/IP context menu using the steps in B.1.2.1, "Accessing the TCP/IP configuration" on page 792.
2. Select **New Interface** from the context menu (Figure 447). Depending on your version and release, you may see the next selection menu Local Area Network, Wide Area Network, or Circuitless. Click **Local Area Network**. You should now see the first window of the TCP/IP wizard interface. Click **Next**. The New TCP/IP Interface Type window appears (Figure 448 on page 794).

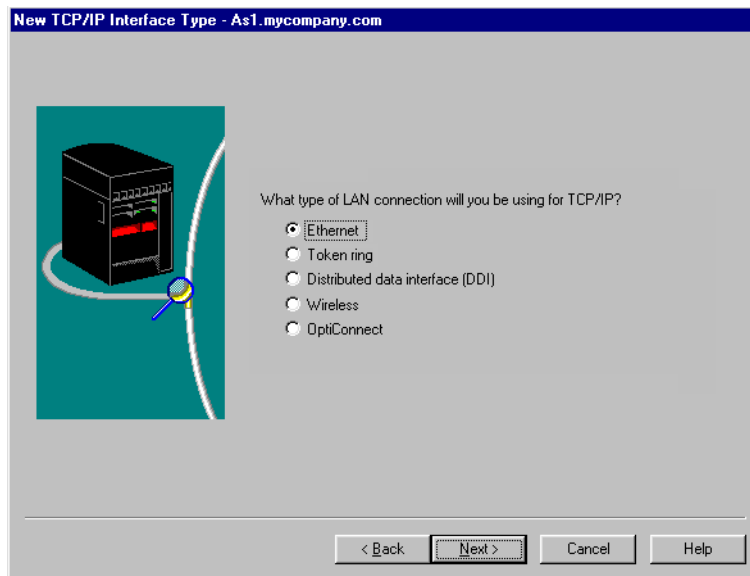


Figure 448. Select Interface Type

3. Select the type of connection you will define for TCP/IP (Ethernet, Token-Ring). In our example, we selected Ethernet. If you select Token-Ring, you may see some different parameters to define. Click **Next**. The New TCP/IP Interface Resource window appears (Figure 449).

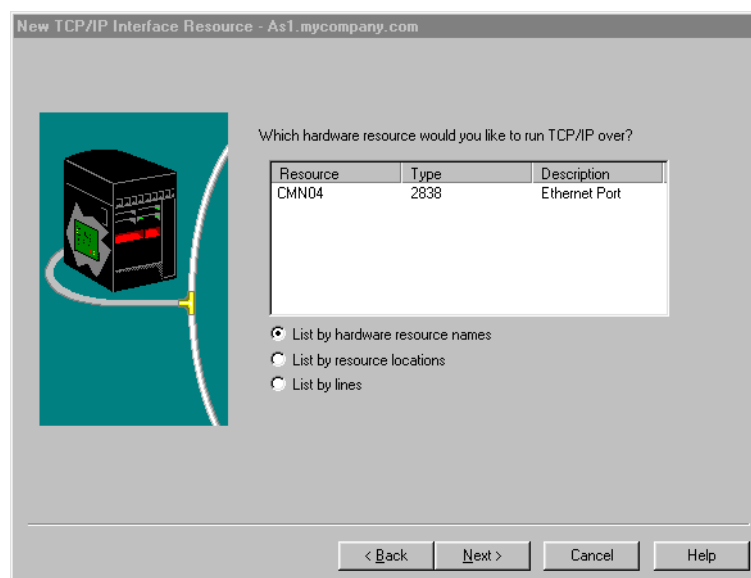


Figure 449. Select Hardware Resource

4. The New TCP/IP Interface Resource window shows all the hardware on your system that matches your type selection. In our example, we have one Ethernet adapter (CMN04). You should use the buttons on the window to determine the location of the adapter. You can also use the buttons to list communication lines that are currently defined. Right-click the hardware resource you want to configure. Click **Next**. The Choosing a Line window appears (Figure 450) if a line is already defined for the hardware resource you selected. Go to the next step. The Creating a New Line Description window

appears (Figure 451) if there are not any lines defined using the selected resource. Go to step 6.

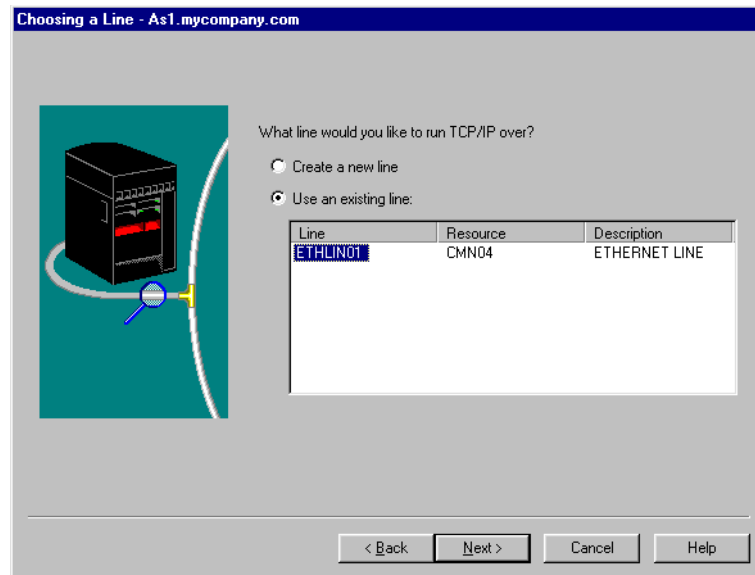


Figure 450. Choosing a Line

5. To configure a TCP/IP interface on an existing line, click **Use an existing line**, and select the line to use from the list provided. Click **Next**. Then go to step 8. To create a new line, click **Create a new line**, and click **Next**. Then, continue with step 6.

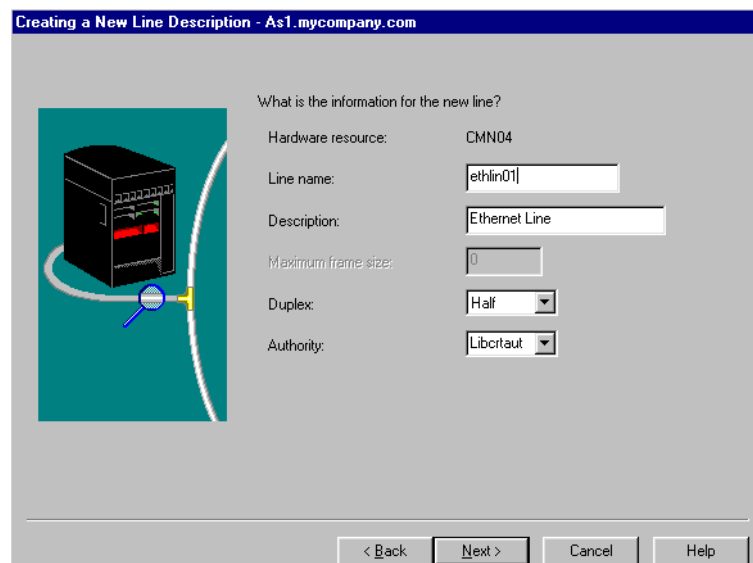


Figure 451. Creating a New Line Description

6. Enter a Name and a Description for the new line. Select the appropriate values for Duplex and Authority based on your environment. The Help button provides additional information to assist you in determining your correct values. The Duplex value is based on the type of network hardware you are using to construct your physical LAN. Click **Next**. The Ethernet Line Characteristics window appears (Figure 452 on page 796).

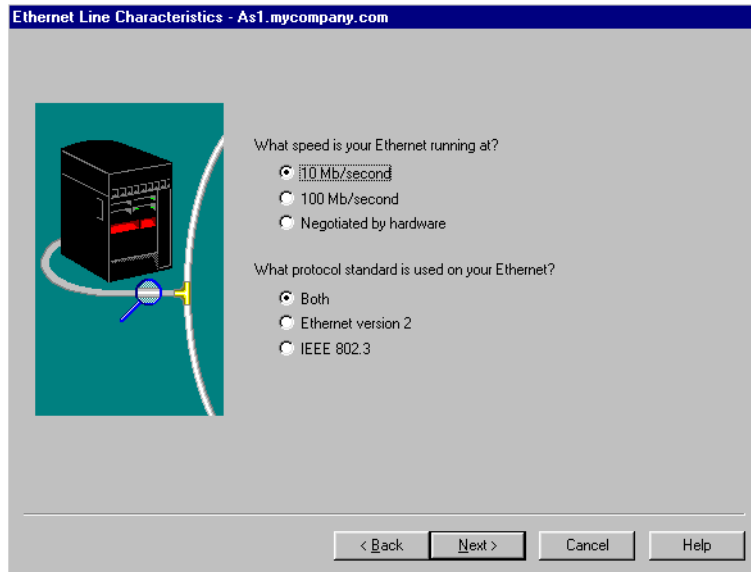


Figure 452. Ethernet Line Characteristics

7. Select the speed at which your LAN is running. Select the protocol standards that you want to support on this adapter. Click **Next**. The TCP/IP Interface Settings window appears (Figure 453).

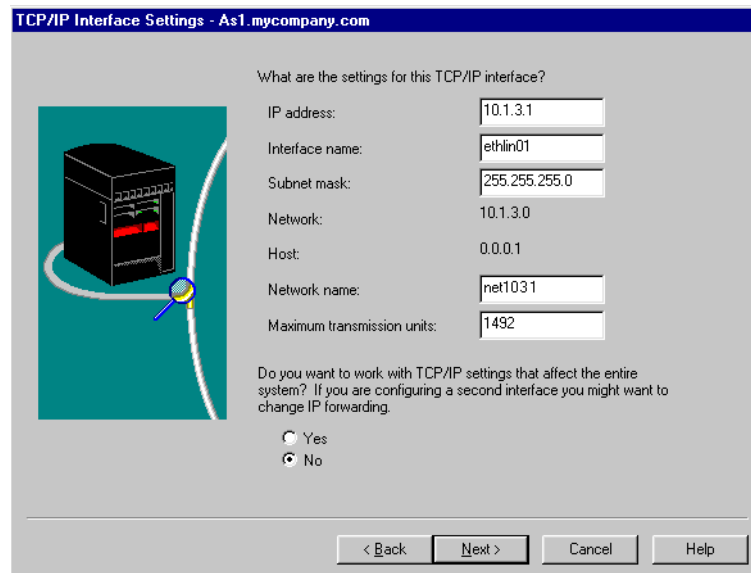


Figure 453. TCP/IP Interface Settings

8. The TCP/IP Interface Settings window (Figure 453) allows you to assign an IP address to your network adapter. Enter the IP address, the Interface name (we used line name), and the Subnet mask for this IP address.

For the IP address and Subnet mask parameter, you specify the value provided by the LAN administrator or Internet Service Provider (ISP). For the IP Address and Subnet Mask, the system does a “logical AND” to determine the Network and Host values displayed in the window. The subnet

mask and the IP address enable IP protocol to determine where to send the data it receives.

Network name specifies the name of the network for which you are defining interfaces and routes for the given network address.

The *Maximum transmission unit (MTU)* specifies the maximum size (in bytes) of IP datagram that you can send on this interface. The maximum size specified for a particular route should not be larger than the smallest MTU that is supported by any router or gateway in that route. If the MTU size is larger than the smallest MTU in the route, the router with the small MTU will fragment the packet. This can increase the traffic on the segment and lead to performance degradation. The Help button provides additional information about MTU.

After you specify all the values, click **Next**. The TCP/IP Routing window displays (Figure 454).

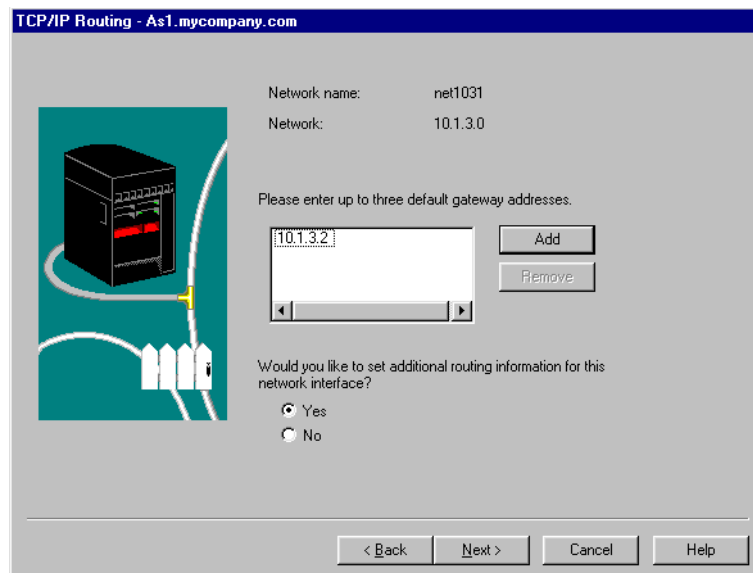


Figure 454. TCP/IP Routing

9. The window in Figure 454 is where you list the gateways to which this route directly connects. A *gateway* is a piece of hardware that connects two or more network segments. It is often called a *router*. You can define up to three gateway addresses. If your AS/400 system is only attached to a single network, then you do not need to specify any gateway addresses. This is also where you specify additional routing information for this interface. This may be used for load balancing or to define multiple routes for backup purposes. Click the **Yes** button to configure additional route information. Click **Next**. The TCP/IP Routing window (Figure 455 on page 798) displays.

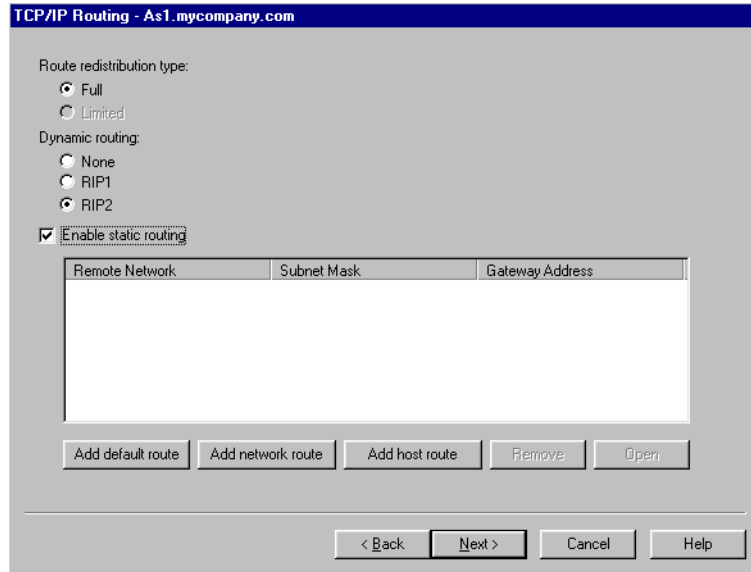


Figure 455. TCP/IP Routing additional information

10. The TCP/IP routing additional information window allows you to specify if these routes should be published to the network using RIP1 or RIP2. You can also define default routes, network routes, and routes to a specific host. Click the appropriate button to add the required routes. In this example, we clicked **Add default route**. The Add Default Route window (Figure 456) appears.

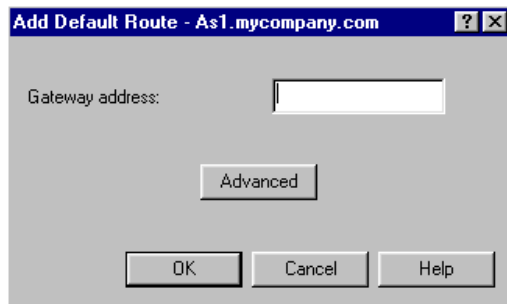


Figure 456. Add Default Route

11. Each of the Add route windows has an Advanced button. Specify the gateway address. Click the **Advanced** button. The Advanced Routing Settings window (Figure 457) appears.

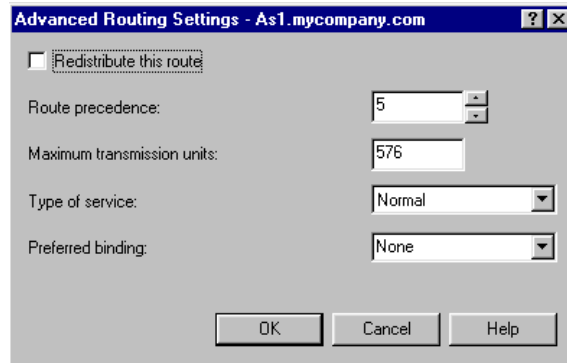


Figure 457. Advanced Routing Settings

12. The Advanced Routing Settings window allows you to specify information about the route. If you leave Route precedence set to 5, then route selection will work as it has always worked. If you set the route precedence to a value of less than 5, this route will not be a preferred route to the destination network. If the route precedence is set to a value greater than 5, then the route will be considered as a preferred route to the destination network. Or, you may have multiple interfaces defined to the same network, and you have multiple routes defined using the interfaces, and the route precedence of these routes are set to the same value greater than 5. In this case, the TCP/IP traffic will be balanced across all the interfaces with routes defined.

Set the values that you need, and click **OK**. If you do not need to set any advanced values, click **Cancel**.

When you have added all the route information you need, click **OK** until the TCP/IP Routing window (Figure 455) displays. Click **Next**. The Servers to be Started window (Figure 458) appears.

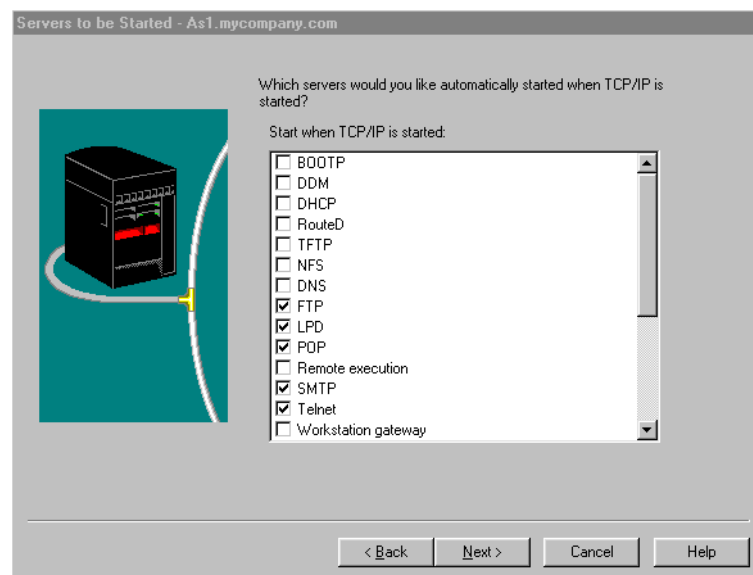


Figure 458. Define Servers to Start when TCP/IP is started

13. From the Servers to be Started window (Figure 458), select all the currently installed servers that you want to start automatically when TCP/IP starts. If

you want to have a particular server automatically started when TCP/IP starts, check the corresponding check box. If you have BOOTP, DHCP, and BOOTP/DHCP servers, only one of them can be checked. After you select all the servers to start, click **Next**. The Start TCP/IP Interface window (Figure 459) appears.

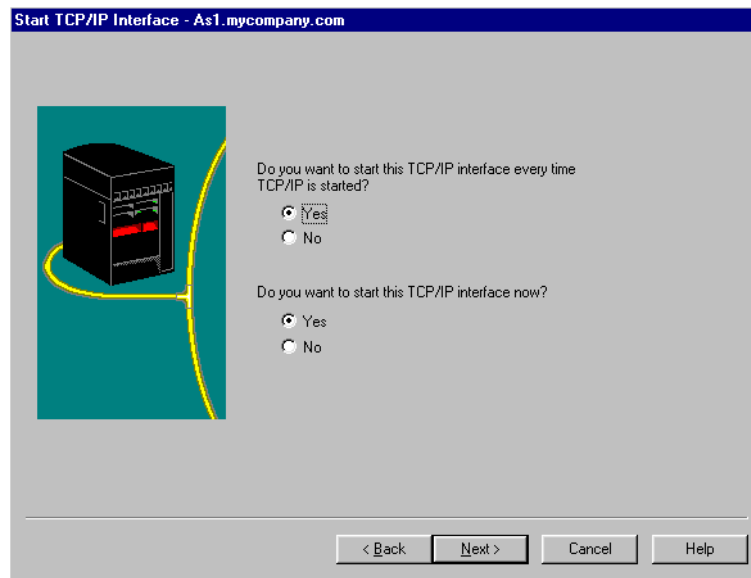


Figure 459. Interface start options

14. From the Start TCP/IP Interface window, you identify whether you want this TCP/IP interface started whenever you start TCP/IP and whether you want this TCP/IP interface to start now. If you choose to start the TCP/IP interface here, the interface begins testing when you click **Next**. After a successful test, the New TCP/IP Interface Summary window (Figure 460) appears.

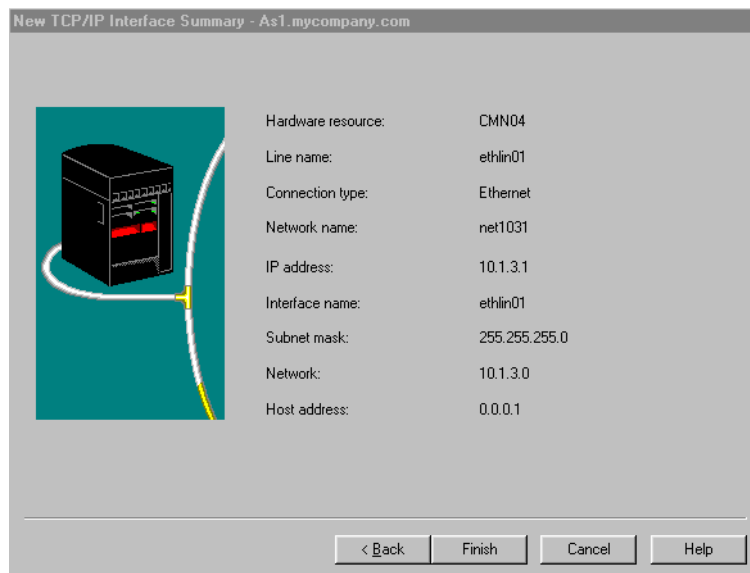


Figure 460. New TCP/IP Interface Summary

15. Verify that all the information displayed is correct. If you need to make changes, click **Back** to go back to the correct window and make your changes. If all the values are correct, click **Finish**.

You have now defined a TCP/IP interface using Operations Navigator.

B.1.2.3 Changing TCP/IP properties

The TCP/IP attributes of the AS/400 system are accessible from Operations Navigator using the properties selection of the context menu. To use the Operations Navigator, perform the following steps:

1. Access the TCP/IP context menu using the steps explained in B.1.2.1, “Accessing the TCP/IP configuration” on page 792.
2. Select **Properties (E)** from the context menu to make detailed changes to the configuration of your TCP/IP interface. Figure 461 on page 802 shows the TCP/IP Properties window. Click **Host Domain Information** to specify the host domain information for your AS/400 TCP/IP communication. Specify the host name, the domain name, and up to three domain name servers. You can also specify the search order and set advanced TCP/IP settings.
 - **Host Name:** Specifies the name for the AS/400 system. You may not always remember a host by its IP address, but you may find it easier to remember hosts by a name. The host name can be combined with the domain name to make a fully qualified name.
 - **Domain Name:** The domain name is a descriptive label for your organization such as your_workplace.com. The two parts of the local domain name are the local domain name and the local host.
 - **Domain Name servers:** List up to three domain server IP address. The system uses the domain servers in the order that you list them. The domain name servers performs host name resolution by translating the host name into an IP address.
 - **Search order:** Specifies whether you want the local host table searched before the domain name server. Figure 461 on page 802 shows the Host Domain Information dialog.

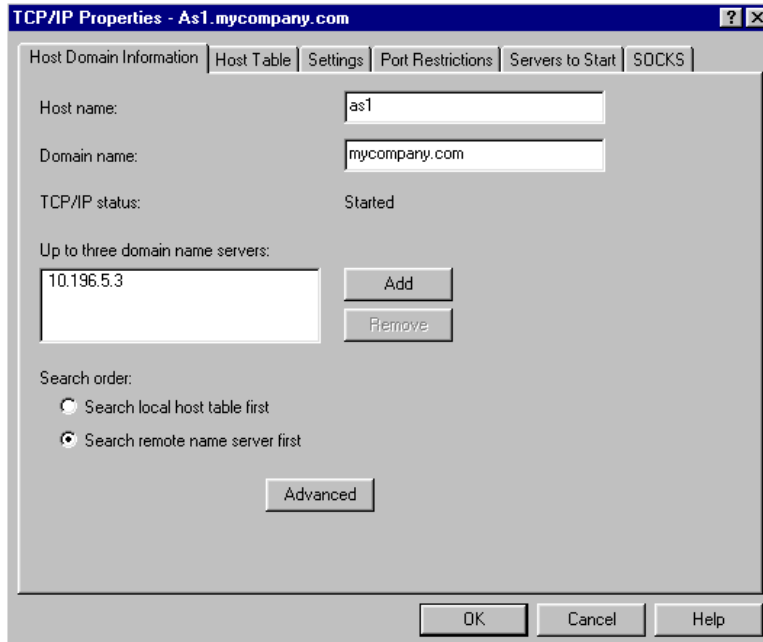


Figure 461. Host Domain Information

3. Click **Advanced** to set additional DNS values. The Advanced Host Domain Information window (Figure 462) appears. The default values shown work in most environments. If you have intermittent trouble resolving names to IP addresses, you may want to increase the number of attempts and the interval between attempts. If these values are set too high, you may experience a long wait time before an “unknown host” message is displayed.

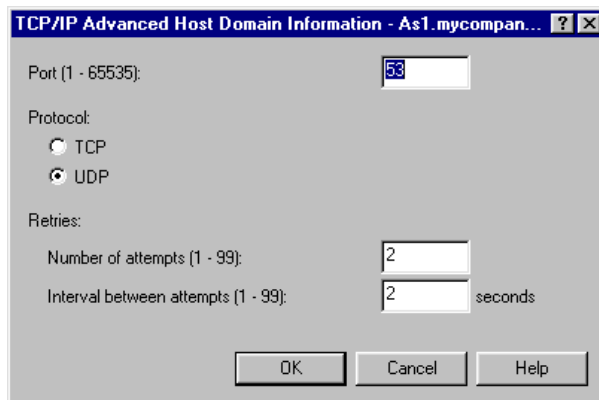


Figure 462. Advanced Host Domain Information settings

4. Click the **Host Table** tab to add and remove host table entries. If you are using the Domain Name System (DNS), you do not necessarily need to add entries here. Figure 463 shows the Host Table dialog.

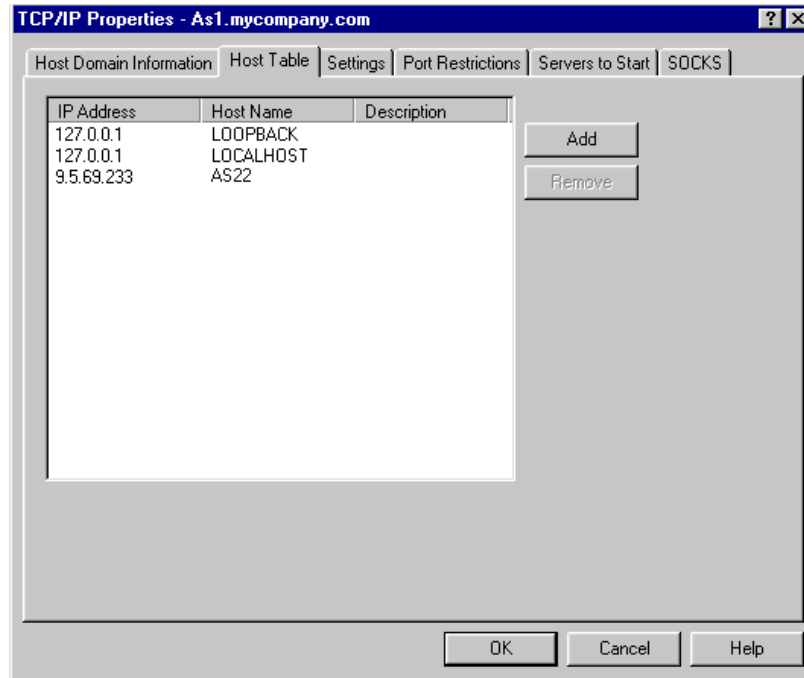


Figure 463. Host Table

5. Click the **Settings** tab to specify IP datagram forwarding, to select a TCP urgent pointer convention, and to enter a TCP keep-alive time. You can also use the settings page to log protocol errors, enable IP source routing and enter a buffer size, time-out, and other values.

IP forwarding specifies whether you want the IP layer to forward IP datagrams between different networks. This specifies whether the IP layer acts as a gateway (router). It allows the AS/400 system to pass IP datagrams that come in one adapter to another adapter.

The TCP keep-alive field specifies the amount of time, in minutes, that TCP waits before sending a probe to the other side of a connection. TCP sends the probe when the connection is otherwise idle, even when there is no data to be sent. Figure 464 on page 804 shows the Settings dialog.

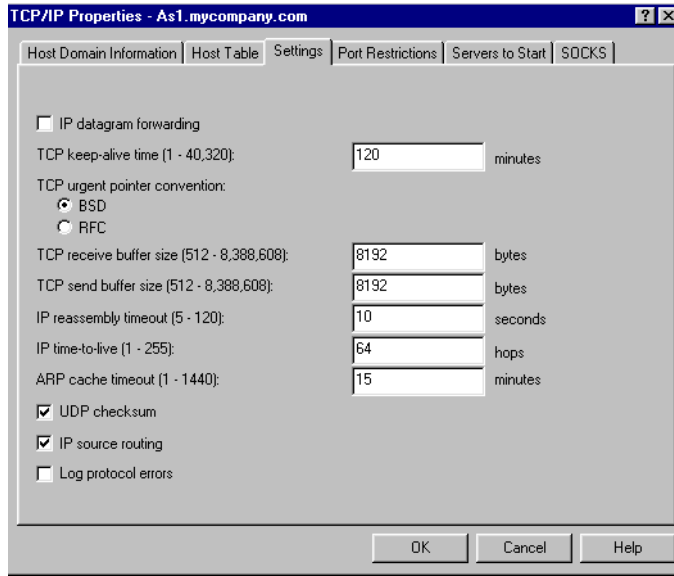


Figure 464. TCP/IP Settings

6. Click the **Port Restriction** tab to limit port use to a user profile name. If you want to restrict a single port, you must specify the same starting and ending port number. Figure 465 shows the Port Restrictions dialog.

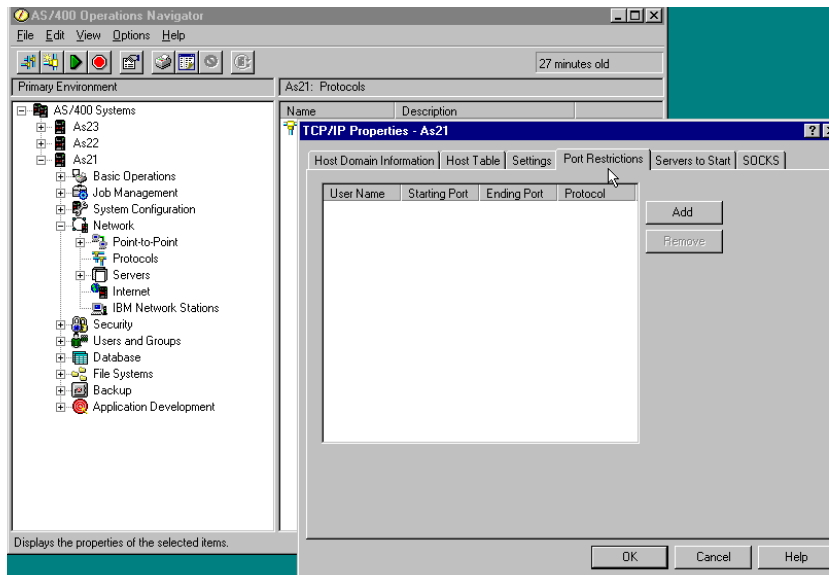


Figure 465. Port restrictions

7. Click the **Servers to Start** tab to select the currently installed servers that you want to start automatically when TCP/IP starts. Check the servers corresponding check box. If you have BOOTP, DHCP, and BOOTP/DHCP servers, only one of them can be checked. Figure 466 shows the Servers to Start dialog.

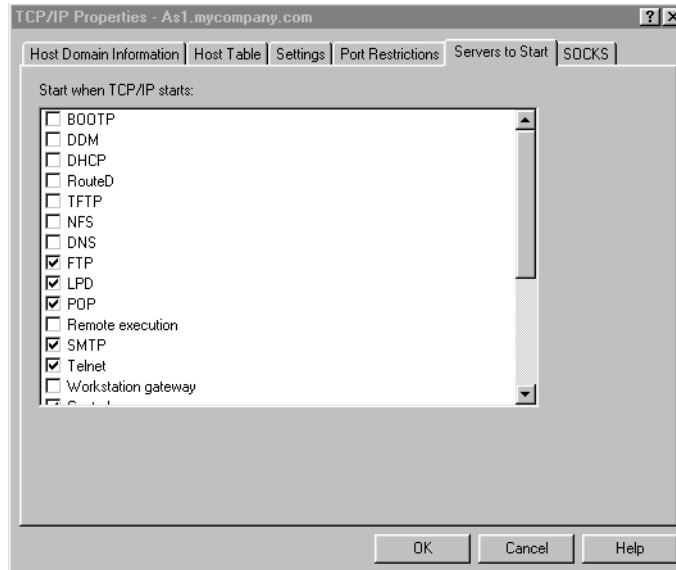


Figure 466. Servers to Start

8. Click the **SOCKS** tab to define the TCP client connection to internal secure networks and to less secure networks. You can define a direct connection to servers in the internal secure network. Users must have *IOSYSCFG special authority to change information on this dialog. Figure 467 shows the SOCKS dialog.

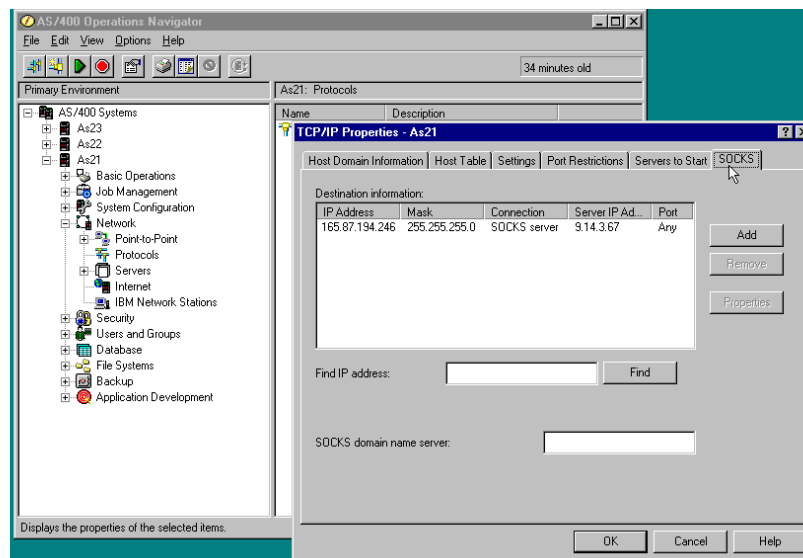


Figure 467. SOCKS

9. After completing changes to the TCP/IP Properties Dialog, click **OK** to save the configuration file and close the window.

B.1.2.4 Configuring host table entries

You must configure host table entries for TCP/IP if you want the users of your AS/400 system to use easily remembered names rather than IP addresses. If you

are using the Domain Name System (DNS), you do not need to configure host table entries.

The host table provides the advantage of not having to remember actual Internet addresses for systems in the network. The host table accomplishes this task by mapping Internet addresses to TCP/IP host names. The local host table on your AS/400 system contains a list of the Internet addresses and related host names for your network.

Before you begin configuring your host table entries for TCP/IP, you need to know the IP addresses of your hosts. You also need to know the host names and descriptions of the hosts that you want to include in the host table.

To configure host table entries for TCP/IP using Operations Navigator, perform the following steps:

1. Select the appropriate TCP/IP window, as follows:
 - a. Start Operations Navigator by clicking **Start ->Programs->IBM Client Access->AS/400 Operations Navigator**. The AS/400 Operations Navigator window appears (Figure 468).
 - b. Double-click the AS/400 Systems icon (A). A list of all the AS/400 systems should appear that can be configured.
 - c. Double-click the AS/400 system you want to configure (B).
 - d. Double-click **Networks** (C).
 - e. Double-click **Protocols** (D).
 - f. Right-click **TCP/IP** to open a context menu (E).
2. Select **Properties** from the context menu (F). Figure 468 shows the context menu - Properties option.

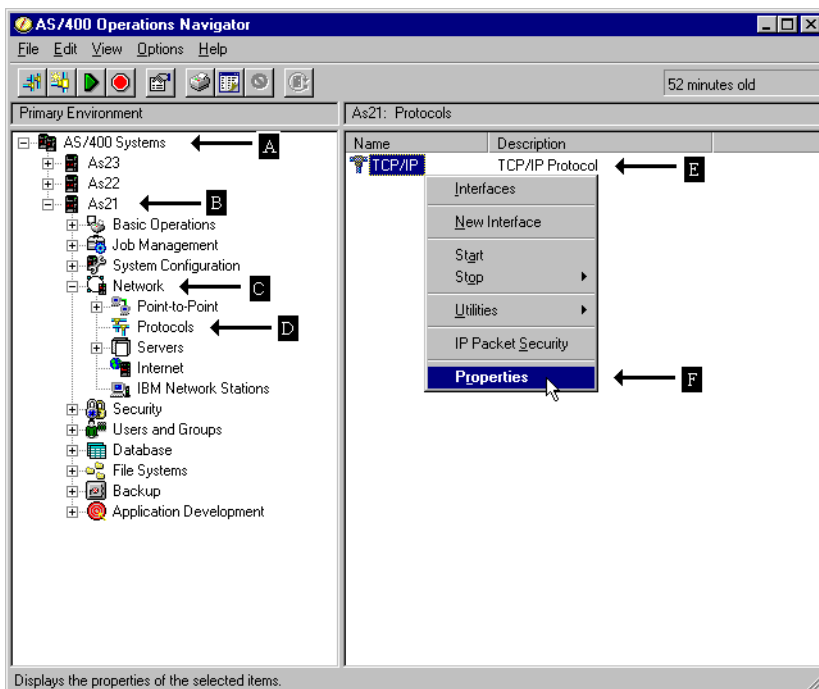


Figure 468. Context menu: Properties

3. Click the **Host table** (A) tab as shown in Figure 469.
4. Click the **Add** (B) button to specify the IP address, hostname, and description of the host that you want to include in the host table. Figure 469 shows the TCP/IP Host Table entry dialog.

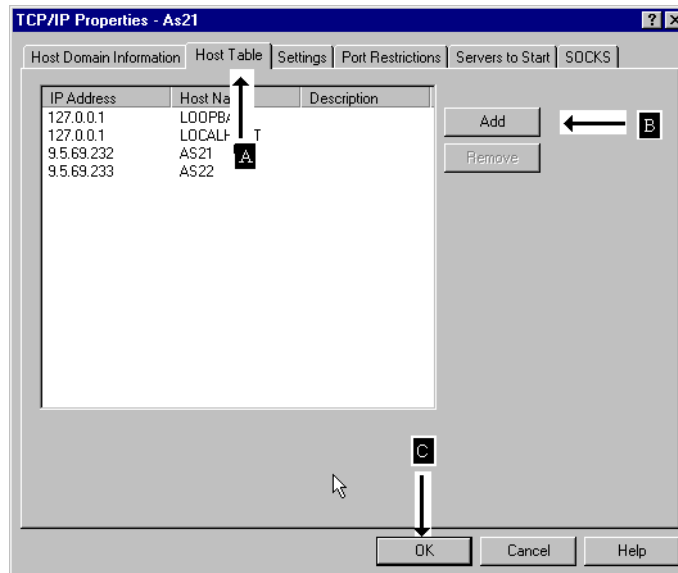


Figure 469. TCP/IP Host Table entry

5. Click **OK** (C) to save the configuration file and close the window.

B.1.2.5 Configuring the domain and host name

You must configure the local domain and host name if you use a remote name server that requires a full domain to resolve an IP address. The local domain name is information that is provided by:

- The network provider
- The local network administrator

This is a “true” intranet if the name is created by the customer.

Within TCP/IP, the primary name associated with your system can have more than one name (your system can have more than one name). It is called your local domain and host name. This is important if you later want to set up e-mail, LPR, and ANYNET. They require the local domain and host name. File transfer and Simple Network Management Protocol use these names, but do not require them.

To configure a local domain and host name for TCP/IP, perform the following steps:

1. Select the appropriate TCP/IP window as follows:
 - a. Start Operations Navigator by clicking **Start->Programs->IBM Client Access->AS/400 operations Navigator**.
 - b. Double-click your AS/400 Systems icon (A). It should give you a list of all the AS/400 systems that you can configure.

- c. Double-click the AS/400 system for which you want to configure a domain and host name (B).
- d. Double-click **Networks** (C).
- e. Double-click **Protocols** (D).
- f. Right-click **TCP/IP** to open a context menu (E).
- g. Select **Properties** from the context menu (F). Figure 470 shows the context menu - Properties option.

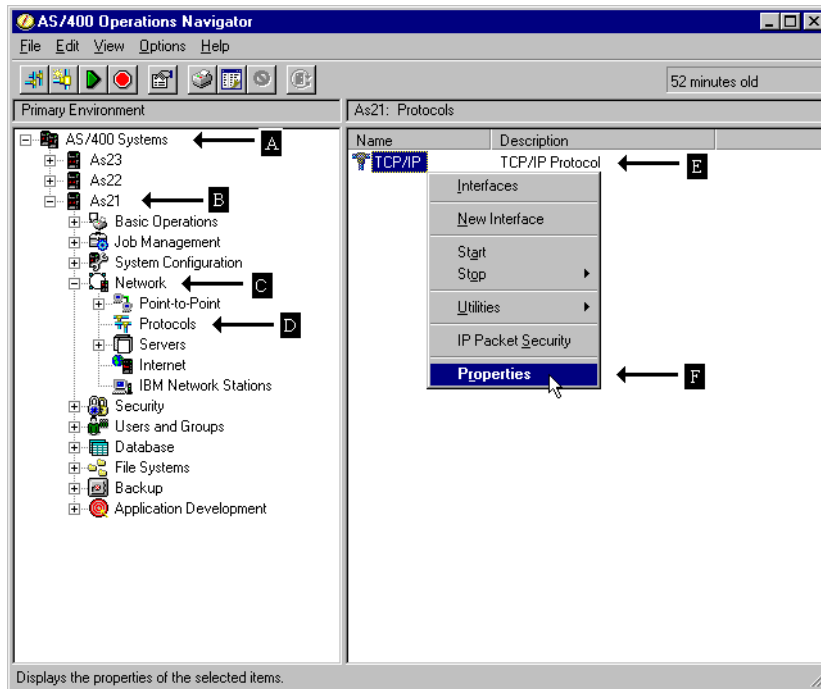


Figure 470. Context menu - Properties

- h. Click the **Host Domain Information** tab (A) as shown in Figure 471.

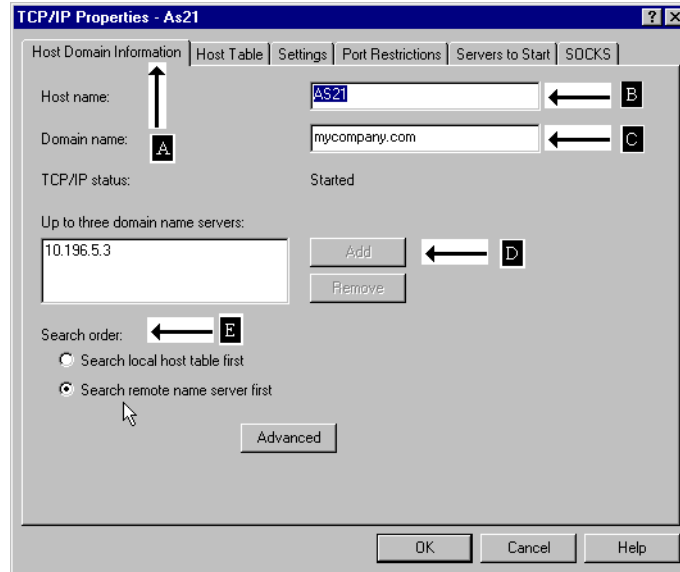


Figure 471. Host Domain Information

2. Specify your host name (B) and domain name (C). You can also select the search order (D), set advanced TCP/IP settings, and specify up to three domain name servers.
3. Click **OK** to save the configuration file.

B.1.2.6 Configuring a TCP/IP route

A network can consist of many interconnected networks. A route must be defined for your system to communicate with a system on another network. If you want to reach remote networks, you need to configure a TCP/IP route for your AS/400 system.

A TCP/IP interface must be defined before defining a route. A TCP/IP interface implicitly defines a direct route. This is because interfaces define a route to a network to which the AS/400 system is directly connected. Routes added using the AS/400 route commands are called *indirect routes* because they define a route to a network to which the AS/400 system is not connected to directly.

The NextHop Internet address for a route definition must exist on a network to which one or more TCP/IP interfaces are connected. The NextHop Internet address usually defines a router or gateway.

Specify the IP address of the router as the default routing entry on the AS/400 system (next hop). This tells the AS/400 system to look for this router if it cannot find a TCP/IP address on its own local network. If you do not configure a TCP/IP route, your AS/400 system cannot reach systems that are on other networks. You may also want to configure a TCP/IP route to give TCP/IP clients access to your AS/400 system.

You do not need to manually configure the routes that tell TCP/IP how to reach the local networks. AS/400 TCP/IP generates these routes automatically from the configuration information for the interfaces every time that TCP/IP starts. Any changes that you make to the routing information take effect immediately.

To configure a TCP/IP route, perform the following steps:

1. Select the appropriate TCP/IP window, as follows:
 - a. Double-click your AS/400 System icon (A) and it should give you a list of all the AS/400 Systems that you are configuring.
 - b. Double-click the AS/400 system that you want to configure a TCP/IP route for (B).
 - c. Double-click **Network** (C).
 - d. Double-click **Protocols** (D).
 - e. Right-click **TCP/IP** to open a context menu (E). Select **New Interface** (F). Figure 472 shows the context menu - New Interface option.

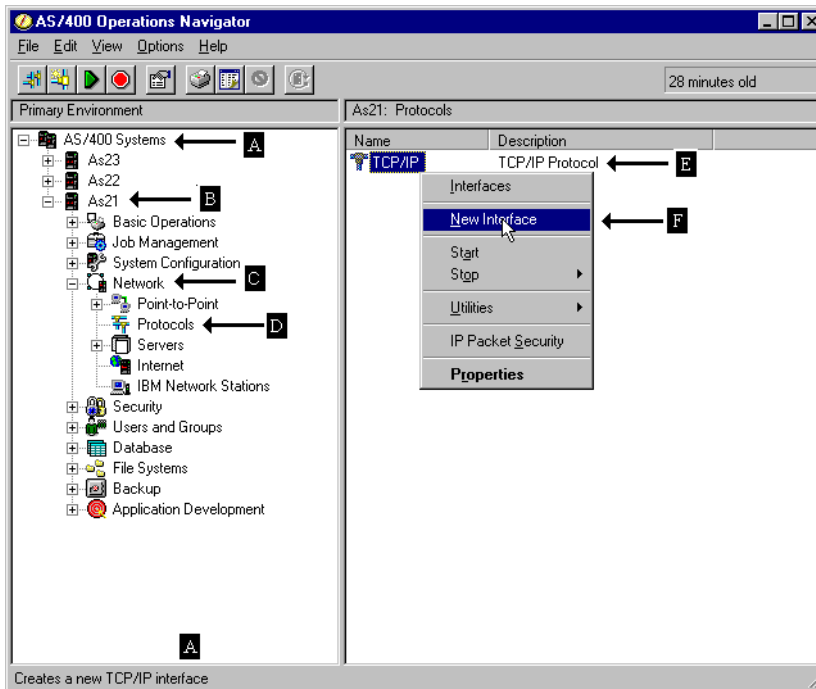


Figure 472. Context menu: New Interface

2. Follow the wizard's instruction to configure your TCP/IP route. Figure 473 shows the first window of the TCP/IP Interface wizard.

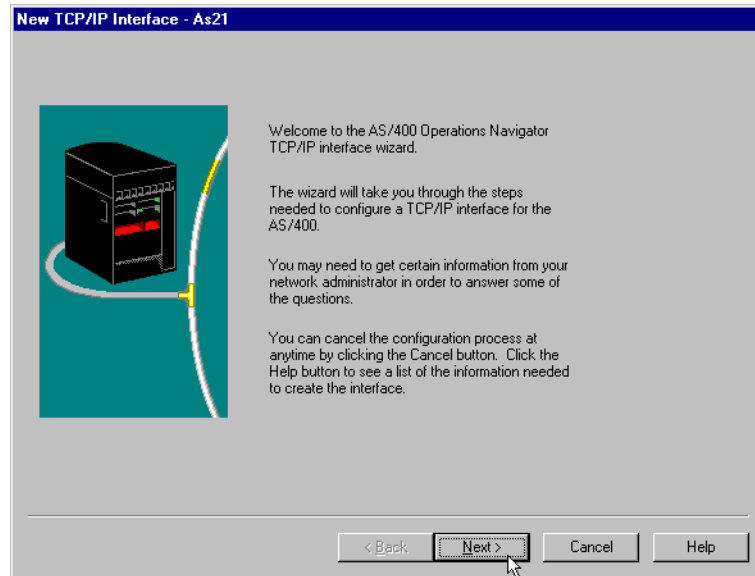


Figure 473. TCP/IP Wizard Interface

B.1.2.7 Starting and stopping TCP/IP

Starting TCP/IP initializes and activates the TCP/IP process, starts the TCP/IP interfaces, and starts the TCP/IP server jobs. TCP/IP must be started before any TCP/IP process can be performed on the AS/400 system. Starting TCP/IP only starts the TCP/IP application jobs that have the AUTOSTART configuration attribute value of *Yes. After starting TCP/IP, the QTCPIP job in the QSYSWRK subsystem is started. The QTCPIP job is used for activating and deactivating TCP/IP interfaces.

When TCP/IP or ANYNET is already active, use the Start TCP/IP Server (STRTCPSVR) command to start additional TCP/IP Application Servers.

Starting TCP/IP

To start TCP/IP, follow this process:

1. Select the appropriate TCP/IP window as follows:
 - a. Double-click the AS/400 system icon (A) in the Operations Navigator tree to give you a list of the AS/400 systems that you are configuring.
 - b. Double-click the AS/400 system that you want to start TCP/IP processing (B).
 - c. Double-click **Network** (C).
 - d. Double-click **Protocol** (D).
 - e. Right-click **TCP/IP** to open the context menu (E).
2. Select **Start** (F) to initialize and activate TCP/IP processing, start TCP/IP interfaces, and start TCP/IP server jobs. Figure 474 on page 812 shows the context menu to start TCP/IP and the Start TCP/IP dialog.

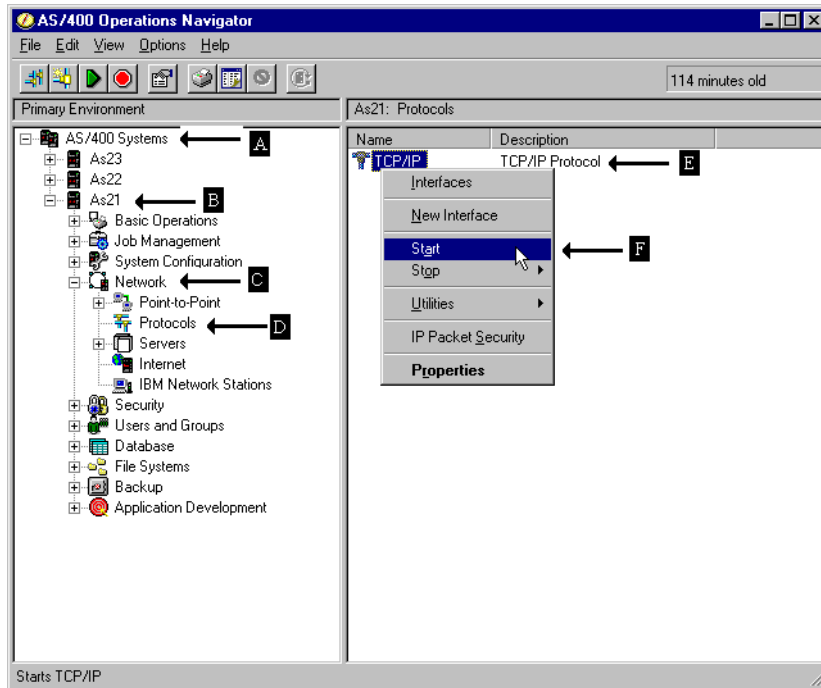


Figure 474. Start TCP/IP

Stopping TCP/IP

Stopping TCP/IP ends all TCP/IP processing, all active TCP/IP interfaces, and all TCP/IP connections on the AS/400 system on which you are working. Unless ENDSVR (*NO) is specified, all TCP/IP server jobs for agents that are currently active in QSYSWRK subsystem are ended. There is no confirmation display shown when stopping TCP/IP, so this should be done with caution. There are two possible values when stopping TCP/IP: controlled and immediately. Follow these steps to stop TCP/IP by using Operations Navigator:

1. Perform the following steps:
 - a. Double-click the AS/400 system icon (A) in the Operations Navigator tree to give you a list of the AS/400 systems that you are configuring
 - b. Double-click the AS/400 system you want to stop TCP/IP processing (B).
 - c. Double-click **Network** (C).
 - d. Double-click **Protocol** (D).
 - e. Right-click **TCP/IP** to open a context menu (E).
2. Select **Stop** (F).
3. Select **Controlled** or **Immediately** (G). Figure 475 shows the Stop TCP/IP dialog.

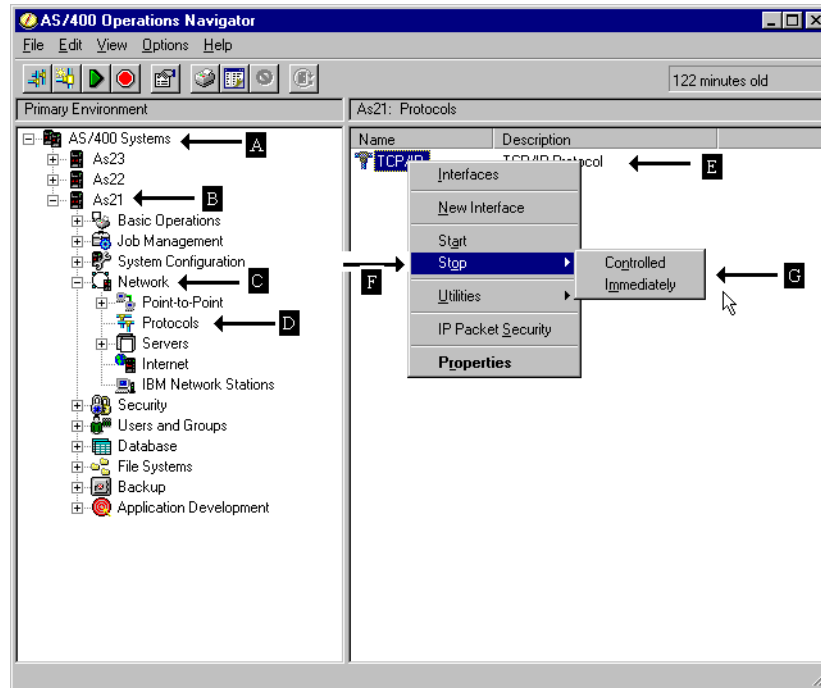


Figure 475. Stop TCP/IP

B.1.2.8 Verifying a TCP/IP connection (ping)

Verifying a network connection (ping) function is one of the best problem determination tools around for quick diagnosis of a problem in your TCP/IP network. Ping tests the TCP/IP connection between a system and the remote system specified on the remote system parameter. It tells you if you can see the host to which you are trying to connect.

When you ping a machine, you send an Internet Control Message Protocol (ICMP) echo request to that machine. A successful reply means that the network's primary transport and communication systems are functioning properly.

To ping a machine using Operations Navigator, perform the following steps:

1. Double-click the AS/400 Systems icon (A), and it should give you a list of all the AS/400 Systems that you can configure.
2. Double-click **Network** (B).
3. Double-click **Protocol** (C).
4. Select **TCP/IP** (D).
5. Right-click **TCP/IP** to open a context menu (E).
6. Select **Utilities** (F).
7. Select **Ping** (G). Figure 476 on page 814 shows the Ping dialog.

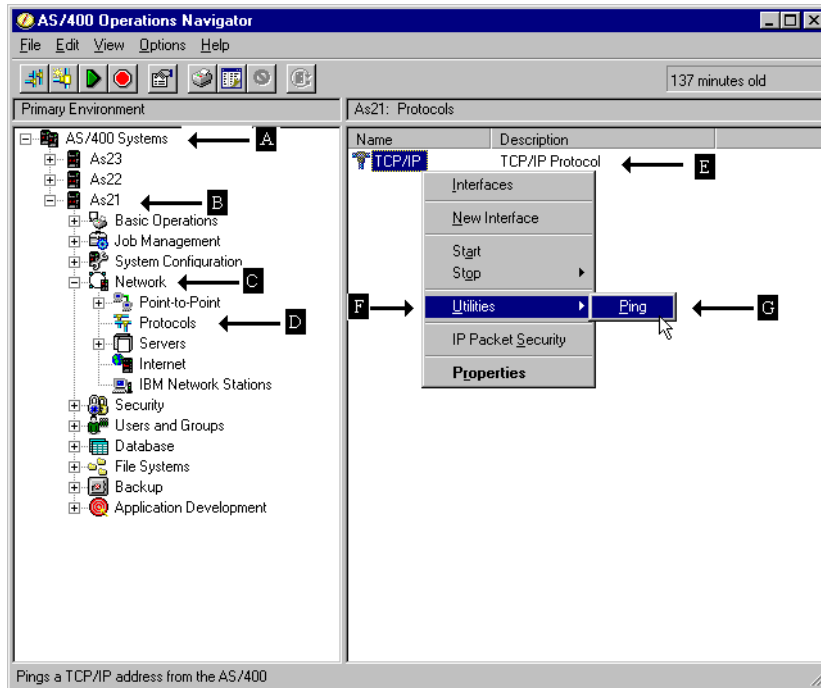


Figure 476. Ping from Operations Navigator

8. As shown in Figure 477, type the IP address or host name of the interface of the host to which you want to test connectivity, and click **Ping Now**. The results of the ping are displayed. Figure 477 shows the Ping from dialog.

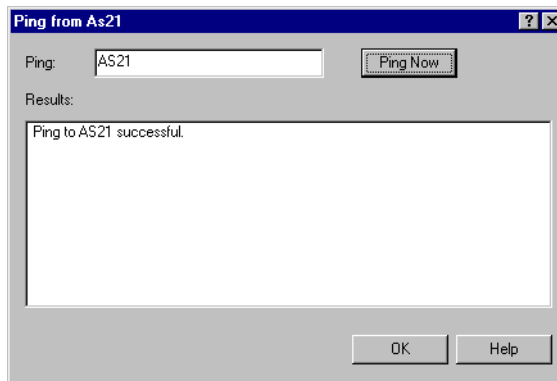


Figure 477. Ping from dialog

B.2 Connecting through a gateway or different network

If you have to connect with a remote host in a different network or subnetwork to the local host (or you use a gateway), it is necessary to configure a route. For example, suppose someone using a PC is attempting to use the TELNET application to start a remote terminal session on this AS/400 system. Obviously, the application on the PC must know the route or path to reach the AS/400 system. However, your AS/400 system must also be able to determine the route back to the PC. If the PC and your AS/400 system are not on the same network, a routing entry must exist both on the PC and on the AS/400 system.

A TCP/IP interface must be defined before defining a route. A TCP/IP interface implicitly defines a direct route. This is because interfaces define a route to a network to which the AS/400 system is directly connected. Routes added using the route commands are called *indirect routes* because they define a route to a network that the AS/400 system is not connected to directly. The NEXTHOP Internet address for a route definition must exist on a network to which one or more TCP/IP interfaces are connected. The NEXTHOP Internet address usually defines a router or gateway. Use the Work with TCP/IP Route Entries display to add route information or to display, change, print, or remove route information.

B.2.1 Scenario

In this scenario, there are two Ethernet networks. We want to connect our first network to the second one. We need a route to know how to go to the 9.5.7.128 network.

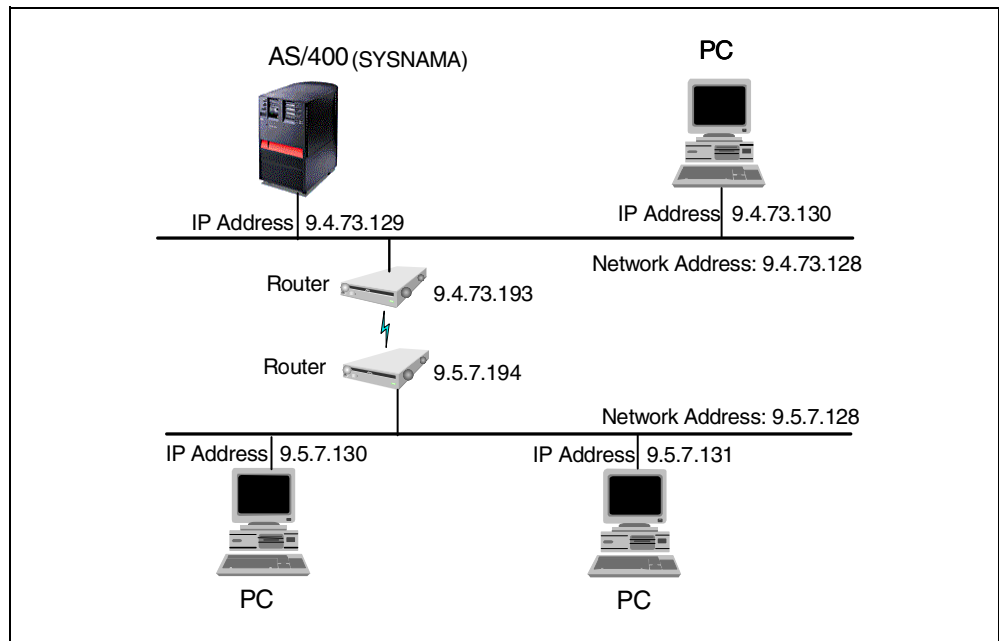


Figure 478. Two Ethernet LANs connected with routers example

B.2.2 Route configuration

To configure a TCP/IP route, perform the following steps:

1. Select option **2** (Work with TCP/IP routes) from the CFGTCP menu, and press Enter. The display shown in Figure 479 on page 816 appears.

```

Work with TCP/IP Routes

Type options, press Enter.
  1=Add  2=Change  4=Remove  5=Display

      Route          Subnet          Next          Preferred
Opt  Destination    Mask           Hop           Interface

      *DFTRROUTE    *NONE          9.4.73.193    *NONE

Bottom

F3=Exit    F5=Refresh  F6=Print list  F11=Display type of service
F12=Cancel F17=Top     F18=Bottom

```

Figure 479. CFGTCP menu display: Work with TCP/IP Routes

2. Select option 1 (Add) to add an entry to the TCP/IP routes. The display shown in Figure 480 appears.

```

Add TCP/IP Route (ADDTCPRTE)

Type choices, press Enter.

Route destination . . . . . > 9.5.7.128  *Note 1
Subnet mask . . . . . > '255.255.255.128'
Type of service . . . . . *NORMAL    *MINDELAY, *MAXTHRPUT...
Next hop . . . . . > '9.4.73.193'
Preferred binding interface . . *NONE
Maximum transmission unit . . . *IFC      576-16388, *IFC
Route metric . . . . . 1          1-16
Route redistribution . . . . . *NO       *NO, *YES
Duplicate route priority . . . . 5         1-10

Bottom

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 480. Add TCP/IP Route (ADDTCPRTE) display

Although there are five parameter values to define a route, you only need to type three of them and assume the others. These values are:

- The route destination: 9.5.7.128
- The subnet mask: 255.255.255.128
- The Internet address of the next system on the route, Next hop: 9.4.73.193

Note 1: Depending on your configuration, the last route destination octet may need to be a “1”.

Appendix C. National language versions on AS/400 systems

This appendix provides a brief overview of some useful information related to national language versions on AS/400 system, such as CCSID, character ID, keyboard ID and so on.

The following tables show national language version feature codes and some of the values associated with each national language version.

Table 62. National language versions on the AS/400 system

National Language Version (NLV)	Feature code	Release	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Albanian	2995	V4R1M0	SQI	500	ALI	697 500	850
Arabic	2954	V3R1M0	ARA	420	CLB	235 420	864
Belgian Dutch	2963		NLB	500	BLI	697 500	850
Belgian English	2909	V4R1M0	ENB	500	BLI	697 500	850
Belgian French	2966		FRB	500	BLI	697 500	850
Brazilian Portuguese	2980		PTB	37	BRB	697 37	850
Bulgarian	2974	V4R1M0	BGR	1025	BGB	1150 1025	855
Canadian French MNCS	2981		FRC	500	CAI	697 500	850
Croatian	2912	V3R1M0	HRV	870	YGI	959 870	852
Czech	2975	V3R1M0	CSY	870	CSB	959 870	852
Danish	2926		DAN	277	DMB	697 277	850
Dutch Netherlands	2923		NLD	37	NEB	697 37	850
English Uppercase	2950		ENP	37	USB	697 37	437
English Uppercase and Lowercase	2924		ENU	37	USB	697 37	437
English Uppercase DBCS (Japanese, No Lowercase)	2938		ENP	5026	JKB	1172 290	897
English Uppercase DBCS (Japanese, Upper/Lowercase)	2938		ENP	5035	JKB	1172 1027	897
English Uppercase and Lowercase DBCS (Traditional Chinese)	2984		CHT	937	TAB	1175 37	1114
English Uppercase and Lowercase DBCS (Simplified Chinese)	2984		CHS	935	RCB	1174 836	1115
English Uppercase and Lowercase DBCS (Korean)	2984		KOR	933	KOB	1173 833	1088
Estonian	2902	V4R1M0	EST	1122	ESB	1307 1122	922
Farsi	2998	V3R1M0	FAR	1097	IRB	1219 1097	1098

(*) Most commonly used PC code page. Actual used code page depends on PC setup.

(**) The language ID for Portuguese and Portuguese MNCS is the same, PTG. Customers using Portuguese MNCS with the PRI keyboard must ensure that the CCSID job attribute is set to 500.

National Language Version (NLV)	Feature code	Release	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Finnish	2925		FIN	278	FNB	697 278	850
French	2928		FRA	297	FAB	697 297	850
French MNCS	2940		FRS	500	SFI	697 500	850
German	2929		DEU	273	AGB	697 273	850
German MNCS	2939		DES	500	AGI	697 500	850
Greek	2957		ELL	875	GNB	925 875	869
Hebrew	2961	V3R1M0	HEB	424	NCB	941 424	862
Hungarian	2976	V3R1M0	HUN	870	HNB	959 870	852
Icelandic	2958		ISL	871	ICB	697 871	850
Italian	2932		ITA	280	ITB	697 280	850
Italian MNCS	2942		ITS	500	ITI	697 500	850
Japanese (Katakana) DBCS	2962		JPN	5026	JKB	1172 290	897
Korean DBCS	2986		KOR	933	KOB	1173 833	1088
Laotian	2906	V4R2M0	LAO	1132	LAB	1341 1132	1133
Latvian	2904	V4R1M0	LVA	1112	LVB	1305 1112	921
Lithuanian	2903	V4R1M0	LTU	1112	LTB	1305 1112	921
Macedonian	2913	V4R1M0	MKD	1025	MKB	1150 1025	855
Norwegian	2933		NON	277	NWB	697 277	850
Polish	2978		PLK	870	PLB	959 870	852
Farsi	2998	V3R1M0	FAR	1097	IRB	1219 1097	1098
Portuguese (**)	2922		PTG	37	PRB	697 37	850
Portuguese MNCS (**)	2996		PTG	500	PRI	697 500	850
Romanian	2992	V4R1M0	ROM	870	RMB	959 870	852
Russian	2979	V3R1M0	RUS	1025	RUB	1150 1025	866
Serbian Cyrillic	2914	V4R1M0	SRB	1025	SQB	1150 1025	855
Simplified Chinese DBCS	2989		CHS	935	RCB	1174 836	903
Slovakian	2994	V3R1M0	SKY	870	SKB	959 870	852
Slovenian	2911	V3R1M0	SLO	870	YGI	959 870	852
Spanish	2931		ESP	284	SPB	697 284	850
Swedish	2937		SVE	278	SWB	697 278	850
Thai	2972	V3R1M0	THA	9030	THB	1279 838	874

(*) Most commonly used PC code page. Actual used code page depends on PC setup.

(**) The language ID for Portuguese and Portuguese MNCS is the same, PTG. Customers using Portuguese MNCS with the PRI keyboard must ensure that the CCSID job attribute is set to 500.

National Language Version (NLV)	Feature code	Release	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Traditional Chinese DBCS (ROC)	2987		CHT	937	TAB	1175 37	1114
Turkish	2956		TRK	1026	TRB	1152 1026	857
Vietnamese	2905	V4R2M0	VNM	1130	VNB	1336 1130	1258

(*) Most commonly used PC code page. Actual used code page depends on PC setup.

(**) The language ID for Portuguese and Portuguese MNCS is the same, PTG. Customers using Portuguese MNCS with the PRI keyboard must ensure that the CCSID job attribute is set to 500.

Table 63. Languages without national language version

Language	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Afrikaans (South Africa)	AFR	37	USB	697 37	437
Australian English (Australia)	ENA	37	USB	697 37	437
Byelorussian (Belarus)	BEL	1025	RUB	1150 1025	1131
Irish Gaelic (Ireland)	GAE	285	UKB	697 285	850
Serbian Latin (Serbia)	SRL	870	YGI	859 870	852
Spanish (Argentina)	ESP	284	SSB	697 284	850
UK English (United Kingdom)	ENG	285	UKB	697 285	850
Ukrainian (Ukraine)	UKR	1123	UAB	1326 1123	1125
Urdu (Pakistan)	URD	918	PKB	1160 918	868

(*) Most commonly used PC code page. Actual used code page depends on PC setup.

Appendix D. Interprocess communication (IPC)

Interprocess communication (IPC) on the AS/400 system is made up of three services:

- Message queues
- Semaphores
- Shared memory

The basic purpose of these services is to provide OS/400 processes a way to communicate with each other through a set of standardized APIs. The three Interprocess Communication services operate using the following IPC objects:

- **Message queues:** This is an IPC mechanism (not to be confused with OS/400 message queue) that allows a job to communicate with other jobs by sending messages to a message queue and receiving messages from a message queue. These message queues are also referred to as OneWorld message queues.
- **Semaphore:** An integer value that can be incremented or decremented. It is used to synchronize two or more processes. A process will wait until a semaphore reaches a certain value before performing an operation.
- **Semaphore set:** A grouping of one or more semaphores.
- **Shared memory:** A piece of storage that is shared between two or more processes. The processes can read or write inside this shared memory.

Although each IPC object provides a specific type of Interprocess Communication, the three services share many similarities. Each service defines a mechanism through which its communication takes place. For message queues, this mechanism is a message queue; for semaphores, it is a semaphore set; and for shared memory, it is a shared memory segment. These mechanisms are identified by a unique positive integer, called respectively, a message queue identifier (msgid), a semaphore identifier (semid), and a shared memory identifier (shmid).

These IPC objects and their identifier values can be viewed using the Applications Development folder within Operations Navigator. Alternatively, you can view the OneWorld specific IPC information by running the Display Interprocess Communication (DSPIPC) command from an AS400 command line, or by selecting the DSPIPC option from the A98OWMNU menu.

IPC resources are based on settings in the jde.ini file. Each instance of OneWorld should have its own address range that does not overlap. Typically 1,000 should be sufficient. However, if there are a large number of call object kernels active, this number may need to be increased. The ranges to use are specified in the JDEIPC section of the jde.ini file, keywords STARTIPCKEYVALUE, MAXNUMBEROF RESOURCES.

Note

These resources are also used for serialization within an instance of OneWorld, but do not provide serialization and lock management among multiple OneWorld instances.

Appendix E. Support structure

This appendix outlines the support available for OneWorld on the AS/400 system and also contains contact information for customers and business partners. The following section describes support available to customers and business partners while E.2, “Business partner support” on page 839, describes support available to business partners only.

E.1 Customer and business partner support

This section describes support and contacts of IBM, J.D. Edwards, Lotus, and Tivoli.

E.1.1 IBM support organizations and contacts

The IBM Worldwide Pre-Sales Technical Support organizations are in place to provide technical pre-sales support. Figure 481 explains the roles and responsibilities of each IBM pre-sales technical support organization (except for Europe, which has combined the missions of Advanced Technical Support and Field Technical Sales Specialists).

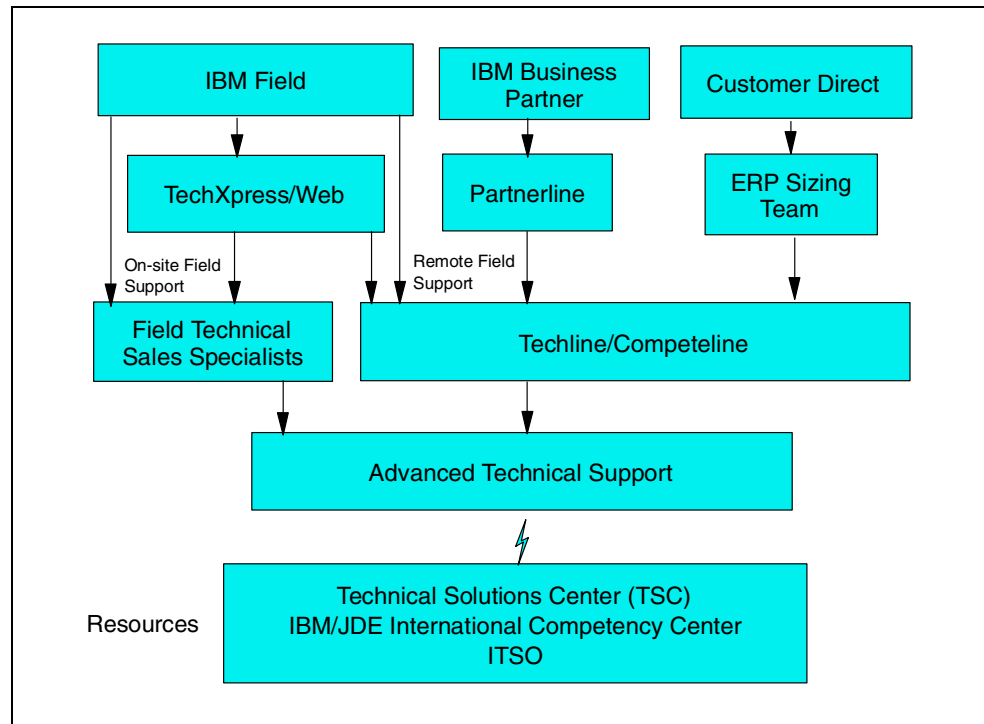


Figure 481. AS/400 J.D. Edwards OneWorld Marketing and Technical Support

Figure 481 displays a typical flow of customer contact from initial inquiry time up to the proposal or final sale time. In addition to this, IBM has established several other organizations, which are described further in the following section.

E.1.1.1 Techline

The mission of the Techline team is to provide technical sales support for ERP opportunities by providing hardware sizing estimates and general marketing

information for specific ERP applications. At this time, the team provides hardware sizing estimates for J.D. Edwards and some other ERP applications. The sizing process consists of gathering requirements using a sizing questionnaire, analyzing the data according to the sizing guidelines, and documenting the hardware recommendation in a sizing deliverable.

Sizing is an iterative process, which is performed many times for prospective or installed customers. During this process, many factors often change, including the customer's requirements, the functionality of the ERP software, and the technology. As a result of this process, it is difficult to measure the accuracy of the sizing. The IBM sizing approach is conservative in anticipation of the changing factors. This approach has contributed to successful installations for many of our customers. If you want to obtain a copy of the sizing questionnaire, you can do this in one of the following ways:

- For self-service, call 800-IBM-4FAX or 1-408-256-4522 (if calling from outside the US). Choose option **2**. Then, order document **8651** "IBM/J.D. Edwards Sizing and Planning Questionnaire".
- Contact the IBM/ERP Pre-Sales Team at: 800-426-0222 or 1-610-251-3969 (if calling from outside the US).
- Send e-mail to: ibmerp@us.ibm.com

E.1.1.2 IBM/J.D. Edwards International Competency Center

The IBM/J.D. Edwards International Competency Center (ICC) in Denver, Colorado, is the second-level support for national competency centers. For example, it provides additional assistance to the North America IBM/ERP Pre-Sales Team, the EMEA IBM/J.D. Edwards Pre-Sales Support Center, and Japan IBM/J.D. Edwards Competency Center. Some of ICCs many tasks include:

- Benchmarking and performance optimization
- Complex sizing support
- Sizing guidelines, sizing guides, and sizing tools
- IBM/J.D. Edwards ICC intranet and Internet site content management

For more information, visit the Web site: <http://www.ibm.com/erp/jdedwards/>

E.1.1.3 EMEA IBM/J.D. Edwards Pre-Sales Support Center

This center provides help with sizings, configurations, technical hints and tips, and proposal inserts. Clients of this center are IBM and J.D. Edwards sales representatives and J.D. Edwards resellers.

E.1.1.4 Japan IBM/J.D. Edwards Competency Center

There is a geographic competency center located in Tokyo, Japan, in place to provide pre-sales and sizing support for opportunities in double-byte language countries.

E.1.1.5 AS/400 Technology Solutions Center

The AS/400 Technology Solutions Center's (TSC) ERP team provides AS/400 system platform-specific technical assistance for marketing and implementation of OneWorld on the AS/400 system. This team of AS/400 system specialists at the TSC assists the IBM/J.D. Edwards Competency Centers, business partners, J.D. Edwards consultants and implementation partners to better understand the AS/400 system as the platform of choice for OneWorld. They assist in the smooth implementation of OneWorld on the AS/400 system by managing AS/400

system-related issues arising during installation, and identifying sources of technical information.

To find out more information about the TSC, you can go to their Web site at:
<http://www.as400.ibm.com/service/bms/jde.htm>

E.1.1.6 IBM Global Services

IBM Global Services is available to work with business partners to ensure large and small companies direct access to innovative, value-added services and solutions, no matter where they do business. They also work with customers of all sizes in a variety of industries all over the world to achieve a competitive advantage.

IBM Global Services can deploy the knowledge and skills of experienced business and IT professionals to help customers build, integrate, and operate IT systems that run their businesses more successfully. They also are designed to help customers install, maintain and derive maximum value from their IT hardware and software and can leverage the power of network computing to help customers reach new markets or do business in totally new ways.

IBM Global Services portfolios include:

- e-business Services
 - IBM e-commerce Services>
 - IBM Enablement Services for e-business
 - IBM Hosted e-business Services
 - IBM Security and Privacy Services
 - IBM e-business Accelerator
- Business Consulting
 - IBM Strategy and Planning Consulting for e-business
 - IBM Business Performance Enhancement Consulting
 - IBM Knowledge Management Consulting
 - IBM e-business Accelerator
- IT Consulting
 - IBM Business Continuity and Recovery Services
 - IBM Network Consulting and Integration
 - IBM Systems Management Consulting and Design
- Business Transformation Services
 - IBM Business Management Services/ERP
 - IBM Global Transformation 2000 Services
 - IBM Business Intelligence Services
 - IBM Supply Chain Management Services
 - IBM Customer Relationship Management Services
 - IBM Enterprise Services for Microsoft Technologies
 - IBM EMU Transition Services
 - IBM Custom Systems Integration Services
- Total Systems Management Services
 - IBM Hardware and Software Support Services
 - IBM Site and Connectivity Services
 - IBM Systems Management and Networking Services
 - IBM High Availability Services

- IBM Business Continuity and Recovery Services
- IBM IT Product Training Services
- Strategic Outsourcing Services
 - IBM Business Process Management Services
 - IBM IT Outsourcing Services
 - IBM Network Outsourcing Services
 - IBM Application Management Services

To find out more information about IBM Global Services, go to their Web site at:
<http://www.ibm.com/services>

E.1.1.7 PC HelpCenter

The IBM Personal Systems Group (PSG) PC HelpCenter provides post-sales technical support (both electronically and by phone). If you are an entitled Business Partner, you're eligible to receive advanced technical support on all PSG hardware products, and problem determination assistance on pre-loaded and selected shrink-wrapped software. Here are the telephone numbers of PC HelpCenters in different geographies:

Table 64 contains a contact list of IBM PC help centers worldwide.

Table 64. IBM PC HelpCenters

Country	Phone
Argentina	11 4717 4357
Australia	800-062-038
Brazil	55 11 889 8986
Canada	800-426-9990
Colombia	800 188 11
Mexico	387 5991 (Mexico City) 01 800 426 1000 (National)
United Kingdom	(+44) 01475 555051
USA	800-IBM-PROD (426-7763)

E.1.1.8 Advanced Technical Support (ATS)

ATS provides world class technical sales support for the design, implementation, and support of solutions through:

- Complex solution sales support
- Skills transfer by subject matter experts
- Expertise in and the creation and maintenance of high-leverage pre-sales tools in the areas of solution assurance, performance tuning, and capacity planning
- Electronic content creation and publishing
- Advanced support to IBM resources requiring more in-depth skills

E.1.1.9 International Technical Support Organization (ITSO)

ITSO develops and transfers technical know-how and materials through IBM Redbooks, workshops, and residencies:

- **Redbooks:** These are practical “cookbooks” written by technical experts worldwide who are developing, implementing, and integrating IBM solutions. You can download these books electronically or enter an IBM subscription through SLSS or via the Web.
- **Workshops:** These are classes that focus on specific skills transfer.
- **Residencies:** These are Internships based on IBM product and implementation plans (contribute to IBM Redbooks and workshops). Internship applications are now online.

Note: You can review and apply for scheduled residencies or workshop sessions. You may also contact your IBM representative or distributor for more information.

You can also sign up on the Web to get announcements of newly available redbooks, residencies, and workshops. Simply go to the IBM Redbooks home page at: <http://www.redbooks.ibm.com>

E.1.1.10 The Centers for IBM e-business Innovation

Work directly with customers to develop and deliver end-to-end e-business solutions. The Centers for IBM e-business Innovation serve as the inbound “brick and mortar” centers, which IBM customers can choose to leverage, bringing together skills in e-business strategy, marketing, interactive design, application development, and business process management.

E.1.1.11 Global e-business integration and e-centers

Design and integrate e-business solutions for IBM customers on leading edge business applications (CRM, SCM, BI, ERP and innovative e-commerce systems). Global e-business integration validates architecture and system design through end-to-end testing, performance verification, security approaches, and design reviews. Engagement is based on opportunity qualification.

E.1.1.12 PartnerWorld

A PartnerWorld Web portal at <http://www-1.ibm.com/partnerworld/pwhome.nsf/my/homepage> is now available as the single point of electronic access for all IBM Business Partners. *PartnerSupport* is the sub-portal for technical support. PartnerWorld is integrating more than 50 different IBM programs for Business Partners including *PartnerInfo*, PartnerCommerce, PartnerRewards and PartnerLine. Business Partners can now access technical support on the Web through PartnerInfo or PartnerSupport.

For information regarding IBM Software Business Partner programs and technical support benefits, go to IBM Business Partner — Software Program Web site at: <http://www.software.ibm.com/partnerweb>

PartnerWorld for Developers

Members of PartnerWorld for Developers receive technical support in the areas of products and information for download and other technical resources to accelerate product development efforts. Developer Support Online offers developers the self-service technical support they need on the Web to help get their products to market faster and includes:

- **Self-service technical support:** Software fixes, FAQs, information maps, sample code, integration starter kits for e-business, and white papers on the latest technologies.

- **Products for download:** The Developer Connection containing a broad selection of e-business tools across 14 platforms, available by download or by CD subscription.
- **Other support and technical resources:** Access to Solution Partnership Centers (SPCs) for technical briefings, hands-on workshops, and cross platform testing/porting facilities and member bulletin boards to allow developers to post questions and trade insights with other developers worldwide.

In addition, PartnerWorld for Developers offers:

- Invitations to technical conferences and events, early release and beta programs, remote S/390 access services, and technology specific consulting.
- *IBM Developer Directions*, a bi-monthly newsletter providing technical information for professional developers and *IBM Developer News*, a biweekly electronic newsletter highlighting the latest information for developers including listing of events, classes, and technical support updates.

PartnerWorld for Developers is located on the Web at:

<http://www.developer.ibm.com/>

E.1.1.13 Specialty centers and support organizations

Many centers that provide specialized help for various platforms and technical areas, such as:

- AS/400 Benchmarking Centers
- Performance Management and Capacity Planning Services
- AS/400 Technical Solutions Center
- Network Computing Division (NCD) Advanced Solution Center (Network Station products)
- Printing Systems Company Remarketer Support
- PS Software Sales and Technical Support

E.1.1.14 IBMLink

IBMLink is the IBM electronic sales manual that provides both public and entitled information. It includes the following electronic support for software through *ServiceLink* in North America or similar services such as *Dial-IBM* in other countries:

- PTF and APAR status tracking
- Defect submissions
- PTF (Program Temporary Fix) downloads
- Preventive service planning
- Technical Q&A databases
- How-to-electronic support through Support Line

E.1.1.15 IBM Web site

The main IBM Web site is located at: <http://www.ibm.com>

E.1.1.16 AS/400 Web site

The main AS/400 Web site is located at: <http://www.as400.ibm.com>

It can also be found at: <http://as400.rochester.ibm.com>

E.1.1.17 IBM technical support telephone numbers

This section contains contact phone numbers in different geographies. Table 65 contains a contact list of IBM Asia Pacific locations.

Table 65. IBM Asia Pacific

Country	PartnerLine	Support line
Australia	(61)1-800-818-090	131-426
Hong Kong	(852)2825-6222	2515-4333
Indonesia	(62)21-523-8377	021-523-8000
Japan	(81) 0088-22-8107	0120-550-508
Korea	(81) 0088-22-8107	82-2-3779-5444
Malaysia	(60) 3-717-7788	603-717-7800
New Zealand	(64) 0800-426-111	04-576-5555
PRC (China)	(800) 810-1166 98	800-810-6677
Philippines	(63) 2-819-2261	819-2271
Singapore	(65) 320-1976	1800-840-9020
Taiwan	(88) 080-011-011	02-776-7776
Thailand	(66) 2-273-4789	2-273-4333

Table 66 contains a contact list of major BM locations in Europe, the Middle East, and Africa.

Table 66. IBM EMEA locations

Country	PartnerLine	Support line
Austria Local	(43) 1 1706 0 0800 201403	1 21145 4400
Belgium	(32) 222 53535	2 718 4333
Denmark	(45) 4523 3277	4596 5050
Finland	(358) 09 4596 400	80014260
France Local	(33) 2 38 55 77 77 0801 801 426	08 01 63 10 20
Germany	(49) 1803 246 256	0511 516 6210
Israel	(972) 03 69 78888	03 69 78555
Italy Local	(39) 039 600 7666 167 015 338	16 782 0094
Netherlands	(31) 20 513 5155	030 285 3939
Norway	(47) 66 99 83 53	66 99 93 00

Country	PartnerLine	Support line
South Africa Local	(27) 11 302 9111 0800 426 426	0800 11 0756
Spain Local	(34) 91 590 5104 901 200 700	901 100 000
Sweden	(46) 0 8793 5008	8793 3000
Switzerland Local	(41) 844 80 30 30 08 44 80 30 30	0800 55 54 54
United Kingdom	(44) 01 256 344500	0345 151516

Country-specific phone numbers are assigned at registration.

Table 67 contains a contact list of IBM locations in North America.

Table 67. IBM North America

Country	PartnerLine	Support line
Canada	800-426-9990	800-IBM-SERV (426-7378)
U.S.	800-426-9990	800-237-5511

Contact IBM Software Developers in North America at 800-627-8363.

Table 68 contains a contact list of IBM locations in Latin America.

Table 68. IBM Latin America

Country	PartnerLine	Support line
Argentina	0800 44 42675	Same
Brazil	0800 78 73 70	Same
Chile	0800 206 666	Same
Colombia	(571) 6230111	(571) 6231300 ext. 1682
Ecuador	(593-2) 565121	Same
Mexico - Mexico City National	(52-5) 2705911 01 800 426 1000 ext. 5911	
Paraguay	5 95 21 206708	Same
Peru	(511) 317 6677	(511) 349 0050
Uruguay	0004054101	Same
Venezuela	800 33426	908 8764 / 908 8295

Note: Product Support phone numbers are used to access Support Line and Passport Advantage support.

E.1.1.18 Defect support and other support information

All problems and defect reports are usually channelled through the Global Support organization. The normal process of defect management and escalation is followed by IBM and J.D. Edwards. Figure 482 shows the typical flow of a customer call from the initial problem inquiry time up to the final problem resolution.

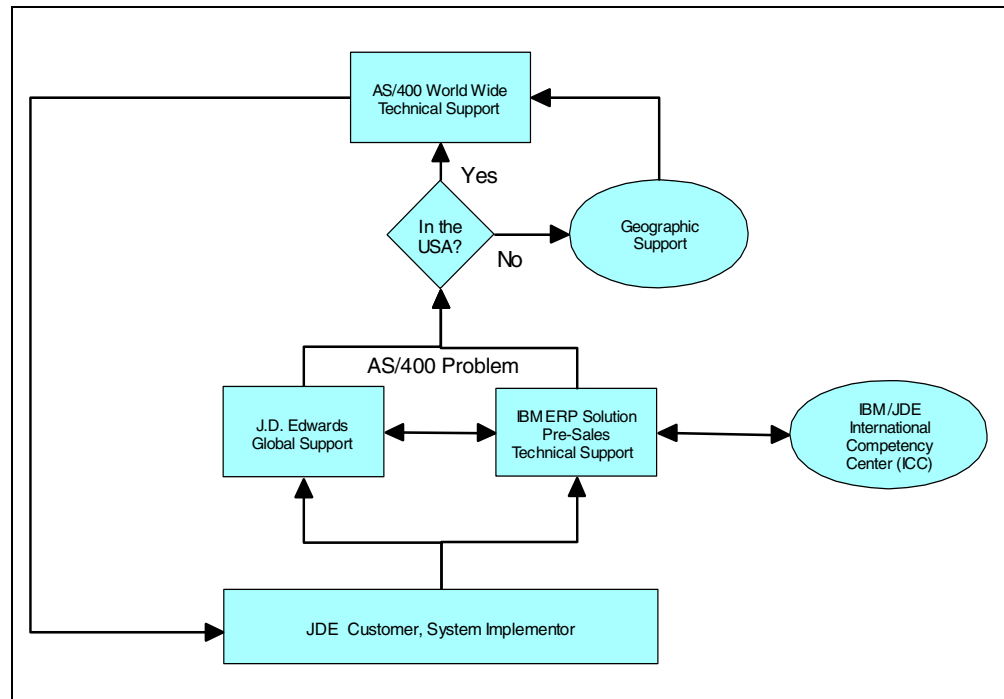


Figure 482. AS/400 and OneWorld defect support

E.1.2 IBM AS/400 Support services

When a customer purchases software from IBM, they receive free support called *program services*. For an additional fee, they can receive access into the AS/400 Support Line. This section goes into more detail regarding the various levels of support.

E.1.2.1 Program services

Program services are available to customers at no additional charge and can be used if an error is suspected in IBM software. Program services is not intended for usage support. AS/400 system specialists help determine and solve the software problem and assist the customer in ordering software fixes (called Program Temporary Fixes or PTFs) from IBM. For program service support, customers can use Electronic Customer Support (or ECS, which is the OS/400 function that allows electronic communications between the AS/400 system and IBM), facsimile, or US mail. Here are the ways to access them:

- **ECS**

 - 1-800-527-8207 (used for ZIP codes 0XXXX-3XXXX)

 - 1-800-327-0949 (used for ZIP codes 4XXXX-9XXXX)

- **Fax**

 - 1-800-288-9584

- **US mail**

IBM Corporation
3605 Highway 52 North
Dept 909
Rochester MN 55901

If a Support Line contract is purchased, customers can receive program services support over the telephone.

E.1.2.2 Support Line

Support Line provides support for usage issues such as problem determination, defect support, installation implementation, and product documentation questions. If a customer desires usage and voice support, they can use Support Line, which is available in both monthly or hourly forms. Depending on the contract, users can call with usage or defect questions 24 hours-a-day, seven days-a-week by calling: 1-800-237-5511.

The following information is required when using Support Line:

- IBM customer number
- AS/400 machine type
- AS/400 serial number
- Support line access number (Support Line hourly customers only)

To streamline support calls, when reporting a problem to IBM, the following information should be available:

- Problem software product name, version, and release level
- Currently installed cumulative PTF number
- Problem symptom
- Message numbers (IDs) with message text and return codes
- The steps needed to recreate the problem (if possible)
- Actions already taken
- AS/400 job log

Note: We highly recommend OneWorld customers to have an IBM support line contract. After contacting AS/400 support, you receive a Problem Management Record (PMR) number for your call. Always keep this number near you in case you need to contact IBM to work on your problem further.

E.1.2.3 Consult Line

When the call goes beyond normal usage or defects, Consult Line is available. Consult Line answers complex questions about systems design planning, interoperability issues, performance and capacity tuning, and LAN/WAN design. It provides telephone access to a group of AS/400 technical experts at the IBM Manufacturing and Development site in Rochester, Minnesota, on an hourly basis. Consult Line has consultants available for such areas as:

- Interpretations
- Recommendations
- Analysis
- Programming and application design
- Performance analysis and capacity planning
- Skill intensive items
- Data recoveries such as tape, diskette, optical platter, and so on

- Explanation of why things are the way they are
- “How to” and “step-by-step” help
- Supported and un-supported products or releases
- OEM devices with the AS/400 system
- PRPQ support (many)

For more information on AS/400 support, please call 1-888-426-4343.

E.1.2.4 AS/400 support Web sites

All AS/400 customers have access to an AS/400 worldwide technical services and support Web sites. Through the support Web sites, a customer can:

- Report a suspected software defect to IBM or ask technical questions directly to support personnel using the Internet if they are an AS/400 Monthly Support Line customer and are registered to use the Internet Facility.
- Order PTFs by the Internet if they are an AS/400 Monthly Support Line customer and are registered to use the Internet Facility.
- Find out the latest in AS/400 news and events.
- Check out the AS/400 Institute, which provides a single source of information to help individuals identify what is required to obtain new, or grow existing, AS/400 skills.
- Link to technical information databases such as:
 - AS/400 Software Knowledge Base
 - AS/400 Software Problem Database (APARS)
 - AS/400 PTF Cover Letters
 - Preventive Service Planning (PSP)
 - Client Access Service Information
 - US Support Center Direct Access Phone Codes
- Link to fixes, drivers, and updates such as:
 - The AS/400 software download library
 - The IBM software download library
 - AS/400 Freeware, Shareware, and Public Domain Software
- Link to technical publications such as:
 - AS/400 Technical Overviews
 - AS/400 On-line Library
 - IBM Redbooks
 - IBM BookManager library
 - AS/400 Magazine (an IBM publication)
 - News400 Magazine
 - Midrange Computing Magazine
 - White papers on data management
 - IBM Technical Journals
- Link to education and certification including:
 - AS/400 Web Builder's Workshop
 - IBM Partners in Education
 - AS/400 Sales School
 - AS/400 Education, Certification, and Publications
 - AS/400 Independent Training Providers

- AS/400 Curriculum Spotlight
- Colleges and Universities providing AS/400 curriculum
- Link to interesting sites, such as:
 - AS/400 Year 2000 support
 - IBM Announcements
 - iSource (IBM Information Service)

To access this information, go to the AS/400 Technical Support Web site:
<http://as400service.rochester.ibm.com>

To use an AS/400 specific search engine, go to the Web site:
<http://www.search400.com/>

E.1.2.5 AS/400 Informational APARs

An AS/400 system informational Authorized Program Analysis Report (APAR) is available for OneWorld that lists the current recommended PTFs and cumulative package level that OneWorld customers should have on their system. Go to:
<http://as400service.rochester.ibm.com/supporthome.nsf/Document/10000035>

Once you reach this site, select **All InfoAPARs by Release**. Using the Search function, type `oneworld`.

OneWorld Informational APARS for specific releases of OS/400 include:

- II11751 - V4R1M0
- II11676 - V4R2M0
- II11677 - V4R3M0
- II11881 - V4R4M0

E.1.2.6 Contact phone numbers

Here is a convenient list of telephone reference numbers available from IBM that provide access to IBM product marketing or support information:

1-800-IBM-4YOU	Provides information about any IBM Solutions or Services.
1-800-IBM-2YOU	Provides information on PC products that you want to order.
1-800-IBM-CALL	This is IBM Direct. Provides information for other systems and software products.
1-800-237-5511	Provides IBM software technical services and support.
1-800-IBM-SERV	Provides IBM hardware technical services and support.
1-800-879-2755	Provides information about ordering IBM publications and media.
1-800-IBM-TEAC(H)	This is IBM Education and Training Customer Service. Provides information about IBM education or courses.

E.1.3 COMMON

COMMON is the world's largest group of IBM and IBM-compatible information technology users. It is an international professional association serving the global community of organizations and individuals who have an interest in IBM-related technology. COMMON provides a member forum for developing proactive

solutions to challenges attributed to technology evolution and business requirements, and provides pre-eminent leadership in education. Members receive unparalleled educational benefits that facilitate the effective use of information systems and enhance professional growth. It also promotes multiple vehicles for communication, an environment that encourages information exchange through interpersonal knowledge sharing, and a level of support otherwise unavailable in the information technology industry. COMMON influences the information industry through its ability to synthesize and communicate member requirements representative of the information technology community. For AS/400 users, COMMON is a way to meet with other companies and users that use the AS/400 system.

Some of the benefits of becoming a COMMON member are:

- **COMMON National Conferences:** Two national conferences are held each year. These five-day events feature more than 900 technical, management, and industry-related sessions, with lectures and labs, plus the COMMON Expo, the industry's largest vendor exhibition. More than 3500 attendees per conference (industry peers, top IBM executives, and major vendors) participate in the ultimate learning experience. You must be a member to attend.
- **COMMON Regional Events:** Recognizing the need for timely knowledge, COMMON stages several Regional Events each year. These events provide the most up-to-date education for IBM users at a local level. Members have access to sessions at various geographic locations all year.
- **Publications:** *COMMONews* is COMMON's official quarterly member newsletter, providing relevant and timely information about the IT industry and the COMMON organization.
- **Requirements System:** The COMMON Requirements Process is being replaced by the new Global Idea Exchange (GLIDE) process. The Requirements Task Force presented the initial design for GLIDE to the COMMON Board of Directors at the fall 1998 conference in Anaheim. The design has also been previewed to COMMON Europe, COMMON Australasia, and IBM. The Requirements System enabled members to submit major concerns about products and services to IBM and midrange vendors 365-days-a-year. More than 1 000 COMMON member requests have become features, functions, products, and services as a result of this process.
- **Top Concerns Process:** Top Concerns focuses on high-level strategic issues such as product direction and overall industry concerns. Members vote annually, and the results are presented to IBM and the midrange industry. The 1997 Top Concerns and the 1997 and 1998 Global Top Concerns can be found on their Web site.
- **COMMON Support Network:** Discover an unparalleled pool of peers and experts (one of the unique features of COMMON is the willingness of members to share solutions and information). This sharing in itself is worth the membership fee!

For more information about membership in COMMON, contact COMMON directly at:

COMMON Headquarters
230 West Monroe, Suite 220
Chicago, IL 60606

Phone: 800-777-6734 or 312-279-0192
Fax: 312-279-0227
E-Mail: common@common.org COMMON Headquarters

To find out more information about COMMON, go to their Web site at:
<http://www.common.org>

E.1.4 J.D. Edwards customer support

Support issues can be logged with J.D. Edwards Worldwide Customer Support via e-mail, telephone, fax, or the Worldwide Web.

E.1.4.1 Web support

Issues can be submitted through the J.D. Edwards Web site at:
<http://www.jdedwards.com>

E.1.4.2 E-mail

Premier maintenance customers sending a support issue by e-mail should use this address: premier_customer_support@jdedwards.com

Standard maintenance customers should send issues by e-mail to the appropriate service provider as shown here:

- Asia Pacific: singapore_customer_support@jdedwards.com
- EMEA: london_customer_support@jdedwards.com
- Japan: japan_customer_support@jdedwards.com
- North and Latin America: denver_customer_support@jdedwards.com

E.1.4.3 Telephone

You can reach J.D. Edwards Worldwide Customer Support centers on the following telephone numbers:

Asia Pacific	65-895-9656
Argentina	001-800-333-1111, then 303-488-4639-9194, then 1-800-784-4260
Australia	1-800-145-200
Brazil	00081-4-550-2779
Chile	1-230-020-5124
China (Beijing, Shanghai)	10-800-650-0099
China (Other Provinces)	108-657, or 108-650. Then follow voice prompts to dial Singapore at 895-9656
Colombia	980-153-560
Ecuador	999-170 then 303-488-4639-9194, then 1-800-784-4260
EMEA	44-1494-682-682
France	33-1-44-74-20-15
Germany	49-6103-762-110
Holland	31-35-548-0222
Hong Kong	800-90-3055
Italy	39-02-2696-7612
Japan	81-3-3265-7199
Malaysia	1-800-80-1557
Mexico	95-800-784-4260

Peru	70 (199 if using a cell phone). Then 303-488-4639-9194. Then 1-800-784-4260.
Spain	44-1494-682-682
U.S. and Canada	1-800-289-2999
Venezuela	800-1-2743

Fax

Support issues can be faxed to the following numbers:

Asia Pacific	65-227-2698
EMEA	44-1494-682698
Japan	81-3-3265-7143
North and Latin America	303-334-4141

During normal business hours, you receive a confirmation within an hour of submitting issues through the Web, e-mail, or fax. At that time, an issue number is assigned. This number, which is assigned immediately when you use the telephone to report an issue, references all activity related to your issue. Outside normal business hours, Premier customers should use the telephone for the most prompt after-hours service.

E.1.4.4 IBM/J.D. Edwards International Competency Center

See E.1.1.2, “IBM/J.D. Edwards International Competency Center” on page 824, and E.1.1.4, “Japan IBM/J.D. Edwards Competency Center” on page 824, for more information.

E.1.4.5 EMEA IBM/J.D. Edwards Pre-Sales Support Center

See E.1.1.3, “EMEA IBM/J.D. Edwards Pre-Sales Support Center” on page 824, for more information.

E.1.4.6 J.D. Edwards Web site

The J.D. Edwards Web site is located at: <http://www.jdedwards.com>

E.1.4.7 J.D. Edwards education and training

J.D. Edwards Custom User Education product consists of two separate components: computer-based training (CBT) and a custom documentation tool. Together these tools provide end users with just-in-time training and ensures that the right training is delivered to the right people at the right time.

In addition, both products are customizable and allow you to incorporate your business-specific information into both the documentation and the CBT. They also support the coexistence of OneWorld and WorldSoftware, as well as multi-language implementations:

- **Custom Documentation Tool:** Custom documentation tool that provides drag-and-drop functionality, which allows you to arrange, build, and customize the J.D. Edwards’ repository of documentation according to your business needs.
- **OneWorld CBT:** OneWorld CBT is a process-oriented training where the majority of lessons cover a series of tasks utilizing an interactive simulation of OneWorld. The lessons include conceptual overviews of important topics, key terms, feedback, hints, access to custom notes, and links to documentation help.

- **WorldSoftware CBT:** WorldSoftware CBT is a series of self-paced, interactive, multimedia-based lessons that are grouped by topic. The students learn the goals and purpose of a specific lesson, and are then lead through the navigation of that specific form or task.

The Web site is located at: <http://www.jdedwards.com/training/>

E.1.5 Lotus and Tivoli contacts

Table 69 contains a contact list of Lotus locations worldwide.

Table 69. Lotus locations in Asia/Pacific

Country	Phone number
United Kingdom	(+44) 01 784 455445
Australia	(+61) 02-9350-7711
Canada	888-263-8714
Caribbean	512-502-5200
China	(+86) 10-6802-8570
France	(+33) 01 4199 5800
Germany	(+49) 180 54 123
Japan	(+81) 03-5496-9847
Korea	(+82) 2-3787-7900
Latin America	512-502-5200
Mexico	512-502-5200
Singapore	(+65) 240-1259
Taiwan	(+886) 2-714-8500
USA	800-553-4270

You can learn more about the Lotus Business Partner Program on the Web at:
<http://www.partners.lotus.com>

You can also investigate the Tivoli Business Partner Program on the Web at:
<http://www.tivoli.com/teamtivoli>

Contact Tivoli North America directly at: 800-TIVOLI8, (848-6548) or 800-237-5511.

E.1.6 QUEST (J.D. Edwards users group)

QUEST's vision is to foster partnership and mutual respect between J.D. Edwards users, existing users groups, and J.D. Edwards through shared knowledge, experience, and professional values. As businesses continue to push for doing more with less, the need for users to get the most out of their software becomes critical. Often, it is difficult to find resources to quickly provide answers to your questions, or a place to get the training and the skills you need. QUEST fulfills this role by helping members to access a knowledgeable network of users

and information on educational opportunities. It also offers members volunteer opportunities. All of this occurs through by partnering with J.D. Edwards to help solve business challenges.

To find out more information about QUEST, go to their Web site at:

<http://www.questnet.org>

E.2 Business partner support

When you need help with marketing situations and opportunities or if you need more technical assistance with OneWorld on the AS/400 system in a pre-sales environment, you may find this section beneficial.

E.2.1 IBM Technical Sales Support

For your Technical Sales Support marketing needs contact PartnerLine, the single point of entry to all key support organizations, including those providing Technical Sales Support consulting services, Product Support Services, and Business Partner programs and services. PartnerLine connects you with Techline, the main interface to hardware and software technical sales support, as well as Competeline (in the U.S. or similar support capabilities in your country) for competitive information and win strategies. You'll receive access to the following types of support for AS/400 systems:

- Product and promotion information
- Product positioning information
- Configuration assistance
- Technical marketing questions
- Initial solution design
- Competitive product and sales strategy information

E.2.2 IBM Solution Assurance

Solution Assurance is a new capability that IBM has provided to help reduce support costs and improve customer satisfaction. It incorporates pre-sale and pre-install technical reviews to minimize the risks and maximize the success of an installation.

Conducting a Solution Assurance Review demonstrate business partner's added value to the customer through a predefined technical evaluation process which:

- Improves solution quality
- Identifies needed skills and services
- Provides a smooth transition from the sale through the installation process

Such activities include:

- Identifying actions required for successful solution implementation
- Ensuring clearly assigned responsibilities
- Anticipating problems or expectations with installation, implementation, and customer satisfaction or expectation
- Leveraging technical support and skills
- Focusing on error-free solution installations

IBM has developed tools for electronic Solution Assurance Reviews (eSARs), product checklists, road maps, and documentation that explain how to proceed with Solution Assurance. These tools and guides help business partners to develop their own internal review procedures and are now available on PartnerInfo (select **Worldwide Technical Support** under the Products category). The Web site and its contents are updated periodically with new information covering additional products and solutions. Questions on Solution Assurance may be directed to PartnerLine.

E.2.3 PartnerInfo

PartnerInfo is an IBM electronic support tool for Business Partners, accessible via Lotus Notes or the Web. You can reach it from IBM PartnerWorld Web site at: <http://www-1.ibm.com/partnerworld/pwhome.nsf/my/homepage>

Click **PartnerInfo**. PartnerInfo includes content and links to all major marketing, sales, and support materials available electronically from IBM (including PC PartnerInfo, Software PartnerInfo, and Networking PartnerInfo).

Select Worldwide Technical Support under “Products” category to gain access to:

- Direct links to all IBM brands technical support sections and content including:
 - Product fixes, drivers and updates
 - FAQs, hints and tips, and other technical databases
- Support Line Family of Services electronic support
- Passport/Advantage software support
- ITSO redbook, workshop, and residency announcements and applications
- Solution Assurance tools and guidelines
- IBM Developer Support Online
- IBM On-Line Technical Support
- IBM Technical Sales Support Information Center
- IBM Technical User Groups

You must be a registered user. If you do not have access to PartnerInfo, contact PartnerLine or your local IBM representative.

E.2.4 IBM Product Support Services

Support Line for Business Partners, an IBM Operational Support Services offering, provides the opportunity to partner with the IBM Support Line team on technical support. This service is similar to the Support Line offering for end users, but has been modified specifically for business partners to assist them in providing support to their customers and to complement their staff. Business partners team with IBM technical support specialists for assistance on installations and turnkey solutions for IBM and selected multi-vendor products. These services are available in most geographies today with enhancements including:

- Cross platform support
- Simplified (single) contract
- Selected multi-vendor support

- Fast and easy access to IBM's technical support specialists
- Availability to all Business Partners. (Some regional limitations may exist.)

Product coverage includes OS/390, VM, VSE, AIX, OS/400, and OS/2 operating systems, along with IBM middleware and networking software products. It also extends to selected vendor products such as Windows NT.

22.6.2.2 Passport Advantage

Passport Advantage is a worldwide software volume licensing program that allows a customer to acquire IBM and Lotus software solutions with the ease of a single contract. Passport Advantage customer support provides easy to access, responsive, cross-platform, cross-product software support from IBM and Lotus, around the clock and around the world. Customer support offered within this program is designed for customers who desire a platform independent software solution. For additional information on Passport Advantage, refer to the Software Support Handbook available through PartnerInfo or at:

<http://www.lotus.com/passportadvantage>

Offerings and their names may vary from country to country. Please contact your local IBM representative or distributor for additional details.

Fee-based post sales support is available to Business Partners in most geographies including:

- **Service Suite Advanced:** Provides prepackaged hardware and software support for customers of IBM products.
- **Support Line for Business Partners:** Helps you support your customers by partnering with our Support Line team.
- **Support Line:** Helps customers with product specific questions about eligible IBM and multi-vendor products.
- **Passport/Advantage:** Helps customers wanting a platform independent software solution.
- **Consult Line (U.S.):** Helps you with more complex questions on design and implementation.

You can be a designated *Named Caller* in your customers support contract and have a seamless interface to IBM.

IBM recommends that your customers sign up for Service Suite or similar complementary offerings to ensure comprehensive support coverage for IBM hardware and software.

Appendix F. Complementary products

This appendix contains a list of J.D. Edwards' business partners, which offer add-on products for OneWorld on the AS/400 system. Their products add specific functionality to the OneWorld suite of products. This listing is ordered by the type of functionality. A contact person from J.D. Edwards is listed for each business partner as well as the company's Web page.

Table 70. Advanced planning and scheduling

Company	Product	J.D. Edwards contact	Company Web site
i2 Technologies	RHYTHM Factory Planner	Brent Dawkins (303) 334-4261	http://www.i2.com
Taylor Manufacturing Systems	TESS	Daryl Latham (303) 334-3476	http://www.taylor-mfgsys.com

Table 71. Analytical applications/datamarts

Company	Product	J.D. Edwards contact	Company Web site
Information Builders Inc.	WorldMART for J.D. Edwards	Jennifer Johanson (303) 334-2134	http://www.ibi.com

Table 72. Applications interoperability

Company	Product	J.D. Edwards contact	Company Web site
Oberon Software Incorporated	Prospero	Jennifer Johanson (303) 334-2134	http://www.oberon.com

Table 73. Computer telephony

Company	Product	J.D. Edwards contact	Company Web site
ComTek	ScreenPhone CTIS	Kristin Izzi (303) 334-5760	http://www.comtek-intl.com

Table 74. Data collection

Company	Product	J.D. Edwards contact	Company Web site
Data Systems International	DcLINK	Daryl Latham (303) 334-3476	http://www.dsionline.com

Table 75. Data warehousing/decision support

Company	Product	J.D. Edwards contact	Company Web site
Cognos Corporation	PowerPlay	Jennifer Johanson (303) 334- 2134	http://www.cognos.com
DataMirror Corporation	Transformation Server/QuickMarts	Jennifer Johanson (303) 334-2134	http://www.data-mirror.com
Information Builders Inc.	SNAPpack and SmartMART for J.D. Edwards	Jennifer Johanson (303) 334-2134	http://www.ibi.com
ShowCase Corporation	Strategy, Sales Analysis, Financial Analysis for J.D. Edwards	Jennifer Johanson (303) 334-2134	http://www.showcasecorp.com

Table 76. Distributed data replication/high availability

Company	Product	J.D. Edwards contact	Company Web site
DataMirror Corp.	Transformation Server, High Availability Suite	Jennifer Johanson (303) 334-2134	http://www.data-mirror.com

Table 77. EAI/interoperability

Company	Product	J.D. Edwards contact	Company Web site
Viewlocity	AMTriX	Jennifer Johanson (303) 334-2134	http://www.viewlocity.com

Table 78. Electronic commerce

Company	Product	J.D. Edwards contact	Company Web site
Harbinger	EDI/400, Trusted Link Enterprise	Daryl Latham (303) 334-3476	http://www.harbin-ger.com

Table 79. Enterprise integration

Company	Product	J.D. Edwards contact	Company Web site
Lotus	Lotus Domino Connector for J.D. Edwards	Kristin Izzi (303) 334-5760	http://www.lotus.com

Table 80. Forms

Company	Product	J.D Edwards contact	Company Web site
AFP Technologies	Formscape	Aimee Robinette, (303) 334- 2171	http://www.formscape.com
Create!Print	Create!Print Form400, Create!Fax	Aimee Robinette, (303) 334- 2171	http://www.form400.com
JetForm	JetForm Design, JetForm Central, JetForm FormFlow 99, JetForm InTempo	Aimee Robinette, (303) 334- 2171	http://www.jetform.com
Optio	OptioDCS	Aimee Robinette, (303) 334- 2171	http://www.optiosoftware.com

Table 81. Hazardous materials management

Company	Product	J.D. Edwards contact	Company Web site
Hazox Corp.	Environmental Mgmt. System	Daryl Latham (303) 334-3476	http://www.hazox.com

Table 82. Imaging

Company	Product	J.D. Edwards contact	Company Web site
Image Integration Systems	Image Management, Image Path	Kristin Izzi (303) 334-5760	http://www.iissys.com
Magellan Software	SpyView, Spylmage	Kristin Izzi (303) 334-5760	http://www.magsoft.com
Optika Imaging	FilePower	Kristin Izzi (303) 334-5760	http://www.optika.com

Table 83. Import/export management

Company	Product	J.D. Edwards contact	Company Web site
Precision Software	TRA/X	Daryl Latham (303) 334-3476	http://www.precision.ie

Table 84. Labor management

Company	Product	J.D. Edwards contact	Company Web site
Data Systems International	DcLINK	Daryl Latham (303) 334-3476	http://www.dsionline.com

Company	Product	J.D. Edwards contact	Company Web site
Kronos	Timekeeper Software	Alicia Gambrell (303) 334-1459	http://www.kronos.com

Table 85. Manufacturing execution system

Company	Product	J.D. Edwards contact	Company Web site
Taylor Manufacturing Systems	TESS	Daryl Latham (303) 334-3476	http://www.taylor mfgsys.com

Table 86. Output/print management

Company	Product	J.D. Edwards contact	Company Web site
Create!Print	Create!Print, Form350 and Create!Fax	Aimee Robinette, (303) 334- 2171	http://www.cpna.com
JetForm	Central, Intempo	Aimee Robinette, (303) 334- 2171	http://www.jetform.com
Loftware Inc.	LLM-WINPDE/Windows NT	Aimee Robinette, (303) 334- 2171	http://www.loftware.com
Optio	OptioDCS	Aimee Robinette, (303) 334- 2171	http://www.optio software.com
Zebra Technologies	BAR-ONE, JetForm Central	Aimee Robinette, (303) 334- 2171	http://www.zebra.com

Table 87. Query/reporting tools

Company	Product	J.D. Edwards contact	Company Web site
Cognos Corporation	Impromptu	Jennifer Johanson (303) 334- 2134	http://www.cognos.com
Information Builders Inc.	SnapPack	Jennifer Johanson (303) 334-2134	http://www.ibi.com
Seagate Software	Crystal Reports	Jennifer Johanson (303) 334- 2134	http://www.seagatesoftware.com

Table 88. System management

Company	Product	J.D. Edwards contact	Company Web site
BMC	BMC Patrol	Kristin Izzi (303) 334-5760	http://www.bmc.com

Table 89. Tax management

Company	Product	J.D. Edwards contact	Company Web site
Vertex, Inc.	SalesTax, Payroll Tax, Quantum for Sales, Quantum for Payroll Tax	Alicia Gambrell, (303) 334-1459	http://www.vertexinc.com

Table 90. Transportation management

Company	Product	J.D. Edwards contact	Company Web site
i2 Technologies	RHYTHM Transportation Management	Brent Dawkins (303) 334-4261	http://www.i2.com

Table 91. Web reporting/e-commerce

Company	Product	J.D. Edwards contact	Company Web site
Information Builders Inc.	WebFOCUS, WorldMART, and SNAPpacks for J.D. Edwards	Jennifer Johanson (303) 334-2134	http://www.ibi.com

Appendix G. Special notices

This publication is intended to help customers, OneWorld consultants, business partners, and IBM specialists who are implementing J.D. Edwards OneWorld on the AS/400 system. The information in this publication is not intended as the specification of any programming interfaces that are provided by J.D. Edwards OneWorld and OS/400. See Appendix H, "Related publications" on page 853, for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

Advanced 36	Advanced Function Printing
Advanced Peer-to-Peer Networking	AFCCU
AFP	AIX
AnyNet	Application System/400
APPN	AS/400
AS/400e	AT
BookManager	C/400
CICS	COBOL/400
CT	Current
DATABASE 2	DataJoiner
DataPropagator	DB2
DB2 Connect	DB2 Universal Database
DRDA	HelpCenter
IBM	IBM.COM
IBMLink	Infoprint
Integrated Language Environment	Intelligent Miner
Intelligent Printer Data Stream	IPDS
Language Environment	Manage. Anything. Anywhere.
Net.Data	Netfinity
Network Station	OfficeVision
OfficeVision/400	Operating System/400
OS/2	OS/390
OS/400	PartnerWorld
RISC System/6000	RPG/400
RS/6000	SAA
S/390	Service Director
SP	SP1
SP2	SQL/400
System/370	System/38
System/390	VisualAge
WebSphere	Wizard
XT	400

The following terms are trademarks of the Lotus Development Corporation in the United States and/or other countries:

Cross-Site	Domino
Lotus	Lotus Notes
Manage. Anything. Anywhere.	NetView
Notes	Passport Advantage
Planet Tivoli	Tivoli
Tivoli Certified	Tivoli Ready
TME	

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix H. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

H.1 IBM Redbooks

For information on ordering these publications see “How to Get ITSO Redbooks” on page 861.

- *AS/400 System Handbook*, GA19-5486
- *AS/400 Printing V*, SG24-2160
- *The System Administrator's Companion to AS/400 Availability and Recovery*, SG24-2161
- *AS/400 - Implementing Windows NT on the Integrated PC Server*, SG24-2164
- *DB2/400 Advanced Database Function*, SG24-4249
- *AS/400 Hierarchical Storage Management*, SG24-4450
- *AS/400 Performance Management V3R6/V3R7*, SG24-4735
- *Inside AS/400 Client Access for Windows 95/NT V3R1M2*, SG24-4748
- *DB2/400: Mastering Data Warehousing Functions*, SG24-5184
- *V4 TCP/IP for AS/400: More Cool Things Than Ever*, SG24-5190
- *AS/400 Client Access Express for Windows: Implementing V4R4M0*, SG24-5191
- *Management Central: A Smart Way to Manage AS/400 Systems*, SG24-5407
- *DB2 UDB for AS/400 Object Relational Support*, SG24-5409
- *Slicing the AS/400 with Logical Partitioning: A How to Guide*, SG24-5439
- *Developing Cross-Platform DB2 Stored Procedures*, SG24-5485

The following redbooks are no longer available in hardcopy format and can only be viewed or downloaded online at: <http://www.redbooks.ibm.com>

At the site, click **Redbooks Online!** Then, on the next page, enter the publication number in the Redbook Search field and click **Submit Search**. Once the results appear, click the title of the book you want to view or download.

- *Speak the Right Language with Your AS/400 System*, SG24-2154
- *AS/400 Client/Server Performance Using Windows Clients*, SG24-4526
- *AS/400 Performance Explorer Tips and Techniques*, SG24-4781
- *J.D. Edwards' OneWorld and IBM Netfinity Implementation Guide*, SG24-5297

H.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at <http://www.redbooks.ibm.com/> for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

H.3 Other resources

These publications are also relevant as further information sources:

- *Ethernet and Token Ring Configuration Guide*, G544-5240
- *ADSTAR Distributed Storage Manager for AS/400 Administrator's Guide*, GC35-0315
- *OptiMover for AS/400*, SC41-0626
- *Client Access for Windows 95/NT - Setup*, SC41-3512
- *Client Access for Windows 95/NT ODBC User's Guide*, SC41-3535
- *OS/400 Server Concepts and Administration*, SC41-3740
- *Backup Recovery and Media Services*, SC41-4345
- *National Language Support*, SC41-5101
- *AS/400 Local Device Configuration*, SC41-5121
- *DB2 for AS/400 Query Manager Use*, SC41-5212
- *OS/400 Security - Reference*, SC41-5302
- *Backup and Recovery*, SC41-5304
- *OS/400 Work Management Guide*, SC41-5306
- *Distributed Data Management*, SC41-5307
- *OS/400 Performance Tools/400*, SC41-5340
- *BEST/1 Capacity Planning Tool*, SC41-5341
- *Backup Recovery and Media Services for AS/400*, SC41-5345
- *Remote Work Station Support*, SC41-5402
- *TCP/IP Configuration and Reference*, SC41-5420
- *TCP/IP Fastpath Setup*, SC41-5430
- *AS/400 Integration with Windows NT Server*, SC41-5439
- *Client Access Express for Windows - Setup*, SC41-5507

- *AS/400 Client Access Express for Windows ODBC User's Guide*, SC41-5509
- *International Application Development*, SC41-5603
- *DB2 for AS/400 SQL Programming*, SC41-5611
- *DB2 for AS/400 SQL Reference Guide*, SC41-5612
- *DB2 for AS/400 Database Programming*, SC41-5701
- *DB2 for AS/400 Query Management Programming*, SC41-5703
- *DB2 Multisystem on AS/400*, SC41-5705
- *Data Management*, SC41-5710
- *Integrated File System Introduction*, SC41-5711
- *DDS Reference*, SC41-5712
- *Printer Device Programming*, SC41-5713
- *CL Reference*, SC41-5722
- *AS/400 Client Access Express Host Servers*, SC41-5740
- *System API Programming*, SC41-5800
- *System API Reference*, SC41-5801
- *DB2 for AS/400 SQL Call Level Interface*, SC41-5806

These J.D. Edwards OneWorld publications are also relevant as further information sources. They may be obtained by attending J.D. Edwards' training courses. For more information, see: <http://www.jdedwards.com>

- *OneWorld Foundation*
- *OneWorld System Administration Guide*
- *OneWorld Configurable Network Computing Implementation*
- *OneWorld Server and Workstation Administration*
- *OneWorld Package Management Guide*
- *OneWorld Application Coexistence*
- *OneWorld Installation Guide (AS/400 Systems)*
- *OneWorld Installation Reference Guide (All Systems)*
- *OneWorld Development Tools*
- *OneWorld Upgrade Guide (AS/400 Systems)*
- *OneWorld Language Installation Guide*
- *OneWorld Enterprise Report Writing*
- *World A7.3 Install Workbook*
- *World A7.3 PTF Install Workbook*
- *WorldSoftware Technical Foundation*

H.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- J.D. Edwards Web site: <http://www.jdedwards.com/>
- J.D. Edwards Education and Training Web site:
<http://www.jdedwards.com/training/>
- QUEST (J.D. Edwards Users Group) Web site: <http://www.questnet.org/>
- J.D. Edwards Knowledge Garden: <https://knowledge.jdedwards.com/>
- AS/400 home page: <http://www.as400.ibm.com>
- IBM AS/400 updates and support for J.D. Edwards OneWorld:
<http://www.as400.ibm.com/service/bms/jde-support.htm>
- AS/400 Worldwide Technical Support: <http://as400service.rochester.ibm.com/>
- OneWorld Informational APARs for specific releases of OS/400:
<http://as400service.rochester.ibm.com/supporthome.nsf/Document/10000035/>

When you reach this page, select **All InfoAPARs by Release**. Select the Search function and type `oneworld`. The following APARs are available:

- II11751 - V4R1M0
 - II11676 - V4R2M0
 - II11677 - V4R3M0
 - II11881 - V4R4M0
- AS/400 software support knowledge base:
<http://as400service.rochester.ibm.com/supporthome.nsf/document/10000051/>
 - AS/400 PartnerWorld for Developers: <http://www.as400.ibm.com/developer/>
 - AS/400 Support Line: <http://www.as.ibm.com/asus/mus62d1.html>
 - AS/400 Consult Line: <http://www.as.ibm.com/asus/consline2.html>
 - AS/400 Customer Benchmark Center:
<http://www.as400.ibm.com/developer/cbc/index.html>
 - AS/400 Education and Certification:
 - <http://www.as400.ibm.com/education/>
 - <http://www.ibm.com/education/certify/>
 - IBM Global Services: <http://www.ibm.com/services/>
 - Rochester Customer Solution Center:
<http://www.as400.ibm.com/service/csc/welcome.htm>
 - AS/400 Technology Solutions Center:
<http://www.as400.ibm.com/service/bms/jde.htm>
 - Custom Technology Center: http://www.as400.ibm.com/service/welcome_3.htm
 - AS/400 Technical Overview: <http://as400service.rochester.ibm.com/>
Expand **Technical Publications** on right side of screen. Then, select **Technical Overviews** at the bottom.
 - AS/400 Beyond Technology: <http://www.as400.ibm.com/beyondtech/>
 - AS/400 University: <http://www.as400.ibm.com/INSTITUTE/internav.htm>
 - AS/400 Windows Integration: <http://www.as400.ibm.com/windowsintegration>

- IBM AS/400 Client Access: <http://www.as400.ibm.com/clientaccess>
 - OBC Section of the Microsoft Universal Data Access Web Site:
<http://www.microsoft.com/data/odbc>
 - AS/400 Information Center, Online Library, and Technical Studio:
<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html>
 - IBM ITSO Information (Internet): <http://www.redbooks.ibm.com/>
 - IBM ITSO Information (IBM Intranet): <http://w3.itso.ibm.com>
 - 29th Street Press (order IBM Redbooks): <http://www.29thstreetpress.com/>
 - Midrange (order IBM Redbooks on the Web): <http://www.midrange.com>
 - Computer Literacy (order IBM Redbooks on the Web):
<http://www.computerliteracy.com/documentation/redbooks>
 - News/400 Magazine: <http://www.news400.com/>
 - The article *64-Bit Computing Made Simple - AS/400 with 64-Bit PowerPC Technology*, can be viewed online at:
<http://iws.as400.ibm.com/WHPAPR/64simple.htm>
 - The AS/400 News Source Midrange Systems:
<http://www.midrangesystems.com/>
 - IBMLink Worldwide IBM Products and Service Information:
<http://www2.ibm.link.ibm.com>
 - IBM online publications: <http://publib.boulder.ibm.com:80>
 - IBM AS/400 Euro home page: <http://www.as400.ibm.com/euro>
 - You can download the V4R3 and V4R4 versions of the SMTRACE and QYPINT libraries from the Web site at:
http://www.as400service.ibm.com/p_dir/pexgui.nsf
 - For information about *Optimal Networks*, a one-to-one relationship network analysis tool that specializes at the application layer, visit the Web site at:
<http://www.optimal.com>
 - For more information about *CACI*, a one-to-many network analysis tool with in-depth information on communications performance, visit the Web site at:
<http://www.caci.com>
 - For more information about *MIL3*, a network planning tool that offers modeling, simulation, and an analysis of the performance of communications networks, computer systems and applications, and distributed systems, visit the Web site at: <http://www.mil3.com>
 - For an online source of IBM publications, visit the Web site at:
<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>
 - A self study lab exercise with a sample OS/400 database, installation instructions, and lab instructions can be downloaded from PartnerWorld for Developers, AS400 (AS/400 Partners in Development), at:
<http://www.as400.ibm.com/developer>
- Select Education->Internet Based Education->Business Intelligence->Piloting DB2 UDB for AS/400 with Operations Navigator.**
- IBM maintains various Informational APARS and documents that detail specific PTFs that are required or recommended to run on the AS/400 system,

as well as other useful information. They can be found at:

<http://as400service.rochester.ibm.com/supporthome.nsf/Document/10000031>

- **ControllIT**, a third-party remote connectivity product provided by Computer Associates, can be accessed at: <http://www.cai.com/products/controlit.htm>
- You can access the latest Client Access Service Packs at:
<http://www.as400.ibm.com/clientaccess/casp.htm>
- For more information on server consolidation, see the IBM Web site at:
<http://www.as400.ibm.com/sc>
- For information on BRMS and HSM, visit the site at:
<http://www.as400.ibm.com/hsmcomp>
- Information regarding OneWorld known issues, tips and techniques, and Software Action Requests (SARs) that may affect your installation processes can be found on the Web at: <http://www.jdedwards.com/solutions/index.asp>
This site requires a valid username and password. You must register with J.D. Edwards to gain access to the site.
- JRun v2.2 can be downloaded in a ZIP file from Live Software's Web site at:
<http://www.livesoftware.com/download>
- For information on support for Windows 2000, visit the Client Access Express Web site at: <http://www.as400.ibm.com/clientaccess/express.htm>
- Visit Client Access Express Toolkit Web page at:
<http://www.as400.ibm.com/clientaccess/toolkit>
- For a complete listing of all the new Client Access Express features, visit the site at: <http://www.as400.ibm.com/clientaccess/express.htm#snapshot>
- For information regarding Operations Navigator and Management Central functions, visit the site at: http://www.as400.ibm.com/oper_nav
- For information regarding Operations Navigator and Management Central functions, visit the site at: <http://www.as400.ibm.com/sftsol/MgmtCentral.htm>
- For information on AS/400 Integration with Windows NT Server, visit the Web site at: <http://www.as400.ibm.com/NT>
- For information on data replication solutions from IBM, visit the site at:
<http://www.software.ibm.com/data/dbtools/datarepl.html>
- For information on data replication solutions from Vision Solutions, Inc., visit the site at: <http://www.visionsolutions.com>
- For information on data replication solutions from Lakeview Technology Inc., visit the site at: <http://www.lakeviewtech.com>
- For information on data replication solutions from DataMirror Corporation, visit the site at: <http://www.datamirror.com>
- Visit the IBM and J.D. Edwards alliance Web site at:
<http://www.ibm.com/erp/jdedwards>
- PartnerInfo is an IBM electronic support tool for Business Partners, accessible on the Web at: <http://www-1.ibm.com/partnerworld/pwhome.nsf/my/homepage>
- PartnerWorld for Developers is located on the Web at:
<http://www.developer.ibm.com/>
- For information about IBM Global Services, go to their Web site at:
<http://www.ibm.com/services>

- Learn about the Lotus Business Partner Program on the Web at:
<http://www.partners.lotus.com>
- For information regarding IBM Software Business Partner programs and technical support benefits, go to IBM Business Partner — Software Program Web site at: <http://www.software.ibm.com/partnerweb>
- For information about QUEST, visit their Web site at: <http://www.questnet.org>
- Learn about the Tivoli Business Partner Program on the Web at:
<http://www.tivoli.com/teamtivoli>
- Visit the IBM AS/400 home page at: <http://as400.rochester.ibm.com>
- Refer to the Software Support Handbook available through PartnerInfo at:
<http://www.lotus.com/passportadvantage>
- Visit the I2 Technologies home page at: <http://www.i2.com>
- Visit the Information Builders Inc. home page at: <http://www.ibi.com>
- Visit the AS/400 Information Center at:
<http://publib.boulder.ibm.com/html/as400/infocenter.html>
- To use an AS/400 specific search engine, go to the Web site:
<http://www.search400.com/>
- To find information about COMMON, go to their Web site at:
<http://www.common.org>
- Visit the Taylor Manufacturing Systems home page at:
<http://www.taylormfgsys.com>
- Visit the Oberon Software Incorporated Web site at: <http://www.oberon.com>
- Visit the Data Systems International Web site at: <http://www.dsionline.com>
- Visit the Cognos Corporation Web site at: <http://www.cognos.com>
- Visit the ShowCase Corporation Web site at: <http://www.showcasecorp.com>
- Visit the Harbinger e-commerce Web site at: <http://www.harbinger.com>
- J.D. Edwards users can access the e-Form (e4m) Solutions Web page at:
<http://www.form400.com>
- Access the JetForm home page at: <http://www.jetform.com>
- Visit the Optio Software Incorporated Web site at:
<http://www.optiosoftware.com>
- Visit the Hazox software solutions Web site at: <http://www.hazox.com>
- Visit the Magellan Software Web site at: <http://www.magsoft.com>
- Visit the Image Integrations Systems Web site at: <http://www.iissys.com>
- Visit the Optika Imaging Web site at: <http://www.optika.com>
- Visit the Precision Software Web site at: <http://www.precision.ie>
- Visit the Loftware Incorporated Web site at: <http://www.loftware.com>
- Access Create!print products at: <http://www.cpna.com>
- Find information about Kronos Timekeeper software at: <http://www.kronos.com>
- Visit the Zebra Technologies home page at: <http://www.zebra.com>
- Visit the Vertex Incorporated home page at: <http://www.vertexinc.com>

- Visit the ComTek International home page at: <http://www.comtek-intl.com>
- Visit the Viewlocity Web site at: <http://www.viewlocity.com>
- Visit the BMC Software Web site at: <http://www.bmc.com>

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** <http://www.redbooks.ibm.com/>

Search for, view, download or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders via e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

This information was current at the time of publication, but is continually subject to change. The latest information for customer may be found at <http://www.redbooks.ibm.com/> and for IBM employees at <http://w3.itso.ibm.com/>.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may also view redbook, residency, and workshop announcements at <http://inews.ibm.com/>.

Glossary

AFP Advanced Function Presentation, Advanced Function Printing

AFPDS Advanced Function Printing Data Stream

ALE Application Link Enabling

APA All Points Addressable

APAR Authorized Program Analysis Report

API Application Programming Interface

APPC Advanced Program to Program Communication

Application Server The server where OneWorld logic is executed. A distributed OneWorld installation can have more than one Application Server.

Application Developers Also known as App Developers. These developers use the OneWorld tools to develop OneWorld business application packages such as Financials, Manufacturing, Distribution, Human Resources, etc.

APPN Advanced Peer-to-Peer Networking

ASCII American National Standard Code for Information Interchange. The character set used by UNIX and Windows.

ASP Auxiliary storage pool

AS/400 Application System/400

ATM Asynchronous Transfer Mode

AWT Abstract Windowing Toolkit

batch job A task or group of tasks you submit for processing that the system treats as a single unit during processing, for example, printing reports and purging files. The computer system performs a batch job with little or no user interaction.

batch processing A method by which the system selects jobs from the job queue, processes them, and sends output to the outqueue. Contrasts with interactive processing.

batch server A server on which OneWorld batch processing requests (also called UBEs) are run instead of on a client, an Application Server, or an Enterprise Server. A batch server typically does not contain a database or run interactive applications.

BBU Battery Backup Unit

BDA See *Business View Design Aid*.

BI Business Intelligence

BLOB Binary large object. A DB2 UDB for AS/400 data type that represents a large stream of binary data treated as a single object.

BRMS Business Recovery and Media Services

browser A client application that translates information sent by the World Wide Web. A client must

use a browser to receive, manipulate, and display World Wide Web information on the desktop. Also known as a Web browser.

BSFN Business Functions code modules written by application developers. This term is used for both individual functions and for the actual C modules, that may contain multiple functions. For example, both b986150.c (the module) and ServerPackageTransfer (the function) will be referred to as business functions in different contexts.

business function An encapsulated set of business rules and logic that can normally be reused by multiple applications. Can execute a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Also contains the APIs that allow them to be called from a form, a database trigger, or a non-OneWorld application. Can be combined with other business functions, forms, event rules, and other components to make up an application. Can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.

business function event rule Encapsulated, reusable business logic created using through event rules rather than C programming. Contrast with embedded event rule. See also *event rule*.

business view Used by OneWorld applications to access data from database tables. A means for selecting specific columns from one or more tables whose data will be used in an application or report. Does not select specific rows and does not contain any physical data. Is strictly a view through which data can be handled.

Business View Design Aid (BDA) This is the OneWorld CASE tool that is used to create and modify business views of data. See also *FDA*, *RDA*, and *TDA*.

category code In user-defined codes, a temporary title for an undefined category. For example, if you are adding a code that designates different sales regions, you can change category code 4 to Sales Region, and define E (East), W (West), N (North), and S (South) as the valid codes. Sometimes referred to as reporting codes.

Call Object Running a business function. Also can be used to mean running a business function on a server. This is the mechanism that allows applications that are running on a client PC to run some set of logic on an Enterprise Server.

CBT Computer Based Training

Central objects Objects that reside in a central location and consist of two parts: the Central objects data source and central C components. The Central objects data source contains OneWorld specifications,

which are stored in a relational database. Central C components contain business function source, header, object, library, and DLL files and are usually stored in directories on the Deployment Server. Together they make up Central objects.

CGI Common Gateway Interface

check-in location The directory structure location for the package and its set of replicated objects, usually `deployment_server\release\path_code\package\packagename`. The sub-directories under this path are where the central C components (source, include, object, library, and DLL file) for business functions are stored.

CISC Complex Instruction Set Computing

CL Control Language

Client Install A mechanism that installs a OneWorld Package from the Deployment Server to a client PC.

client/server A relationship between processes running on separate machines. The server process is a provider of software services. The client is a consumer of those services. Provides a clean separation of functions based on the idea of service. A server can service many clients at the same time and regulate their access to shared resources. There is a many-to-one relationship between clients and a server, respectively. Clients always initiate the dialog by requesting a service, and servers passively wait for requests from clients.

CLP Control Language Program

CNC See *Configurable Network Computing*.

Conference Room Pilot (CRP) The process of installing, configuring and running a small scale model of the production system to work out the details of the final installation.

configurable client engine Allows user flexibility at the interface level. Users can easily move columns, set tabs for different data views, and size grids according to their needs. The configurable client engine also enables the incorporation of Web browsers in addition to the Windows 95- and Windows NT-based interfaces.

Configurable Network Computing Application architecture that allows interactive and batch applications, composed of a single code base, to run across a TCP/IP network of multiple server platforms and SQL databases. Applications consist of reusable business functions and associated data that can be configured across the network dynamically. The overall objective for a businesses is to provide a future-proof environment that enables them to change organizational structures, business processes, and technologies independently of each other.

core The central and foundation systems of J.D. Edwards software, including General Accounting, Accounts Payable, Accounts Receivable, Address

Book, Financial Reporting, Financial Modeling and Allocations, and Back Office.

CPM Continuously Powered Main storage

CQP See *Customer Quality Program*.

CRP See *Conference Room Pilot*.

cumulative release A release containing changes to both system and application code. These releases are cumulative between base releases, for example, B73. 1.2 contains all changes that were done for B73. 1.1. Also known as Cums.

Customer Quality Program (CQP) Basically OneWorld beta sites.

DAL Data Access Layer

DASD Direct Access Storage Device

Data Dictionary OneWorld method for storing and managing data item definitions and specifications. J.D. Edwards has an active Data Dictionary, which means it is accessed at runtime.

data mart Department-level decision support databases. Usually draw their data from an enterprise data warehouse that serves as a source of consolidated and reconciled data from around the organization. Data marts can be either relational or multidimensional databases.

data replication In a replicated environment, multiple copies of data are maintained on multiple machines. There must be a single source that "owns" the data. This ensures that the latest copy of data can be applied to a primary place and replicated as appropriate. This is in contrast to a simple copying of data, where the copy is not maintained from a central location, but exists independently of the source.

data source A specific instance of a database management system running on a computer. Data source management is accomplished through Object Configuration Manager (OCM) and Object Map (OM).

data warehouse Database used for reconciling and consolidating data from multiple databases before it is distributed to data marts for department-level decision support queries and reports. The data warehouse is generally a large relational database residing on a dedicated server between operational databases and the data marts.

data warehousing Involves off-loading operational data sources to target databases that will be used exclusively for decision support (reports and queries). There are a range of decision support environments, including duplicated database, enhanced analysis databases, and enterprise data warehouses.

database A continuously updated collection of all information that a system uses and stores. Make it possible to create, store, index, and cross-reference information online.

DB2 Universal Database for AS/400 (DB2 UDB for AS/400) The integrated database management system of the AS/400 operating system that provides the structure, integrity, and data manipulation components of the relational database model.

database administrator (DBA). The person whose job is to administer a Database Management System (DBMS). There is *less need* for this task on the AS/400 system due to the integrated nature of DB2 UDB for AS/400.

database driver Software that connects an application to a specific database management system.

Database Server A server that stores data. Does not have OneWorld logic.

DB2 UDB for AS/400 See *DB2 Universal Database for AS/400*.

DBCS See *Double Byte Character Set*.

DCE See *distributed computing environment*.

DDL Data Definition Language

DDM Distributed Data Management

DDS Data Description Specification

Deployment Server A file server that contains business function source, header, object, and DLL files, TAM specifications, and packages. This is where Object Librarian goes to get files when they are checked out, and put them when they are checked in. Client Install takes a package from the Deployment Server and deploys it to a client PC.

DHCP Dynamic Host Configuration Protocol

Distributed Computing Environment (DCE)

A set of integrated software services that allows software running on multiple computers to perform in a manner that is seamless and transparent to the end-users. DCE provides security, directory, time, remote procedure calls, and files across computers running on a network.

DLL See *dynamic link library*.

DML Data Manipulation Language

Double Byte Character Set (DBCS) Character sets that require two bytes to identify each character, such as Chinese or Japanese.

DNT Do Not Translate

DSS Decision Support System

DST Dedicated Service Tools

duplicated database A decision support database that contains a straightforward copy of operational data. Advantages involve improved performance for both operational and reporting environments. See also *enhanced analysis database* and *enterprise data warehouse*.

dynamic link library (DLL) A set of program modules that are designed to be invoked from executable files when the executable files are run, without having to be linked to the executable files. Typically contain commonly used functions.

dynamic partitioning The ability to dynamically distribute logic or data to multiple tiers in a client/server architecture.

EBCDIC See *Extended Binary Coded Decimal Interchange Code*.

EC Electronic Commerce

ECS Electronic Customer Support

EDI Electronic Data Interchange

EIS Executive Information System

enhanced analysis database A database containing a subset of operational data. The data on the enhanced analysis database performs calculations and provides summary data to speed generation of reports and query response times. This solution is appropriate when external data must be added to source data, or when historical data is necessary for trend analysis or regulatory reporting. See also *duplicated database* and *enterprise data warehouse*.

enterprise data warehouse A complex solution that involves data from many areas of the enterprise. This environment requires a large relational database (the data warehouse) that is a central repository of enterprise data, which is clean, reconciled, and consolidated. From this repository, data marts retrieve data to provide department-level decisions. See also *duplicated database* and *enhanced analysis database*.

Enterprise Server A Database Server and logic server. Also referred to as a host. See *Database Server*.

Environment The set of TAM files, business functions, and data that are used when running OneWorld. What is in an environment is driven by the path code for the environment, which drives what set of business functions and TAM files are used, and by OCM, which describes where all of the data resides.

ER See *Event Rules*

ERP Enterprise Resource Planning

Event Rules Specifications that describe what OneWorld will do when a given event occurs. Stored in the gbrspec TAM file.

Extended Binary Coded Decimal Interchange Code (EBCDIC)The character set used by the AS/400 system.

External Service Packs OneWorld patches that are delivered to customers.

facility A separate entity within a business for which you want to track costs. For example, a facility may be a warehouse location, job, project, work center, or

branch or plant. Sometimes referred to as a business unit.

A collection (in Home Builder and ECS) of computer language statements or programs that provide a specialized function throughout a system or throughout all integrated systems. For example, DREAM Writer and FASTR are facilities.

fat client A client that uses its own resources, in the form of processor, memory, and disk, to perform substantial amounts of application processing in cooperation with a server.

FDA See *Form Design Aid*.

FDDI Fiber Distributed Data Interface

find/browse Type of form used to: 1) Search, view, and select multiple records in a detail area. 2) Delete records. 3) Exit to another form. 4) Serve as an entry point for most applications.

firewall Set of technologies that allows an enterprise to test, filter, and route all incoming messages. Are used to keep an enterprise secure.

fix/inspect A type of form used to view, add, or modify existing records. Has no detail area.

form An element of OneWorld's graphical user interface that contains controls by which a user can interact with an application. Forms allow the user to input, select, and view information. A OneWorld application may contain multiple forms. In Microsoft Windows terminology, a form is known as a dialog box.

Form Design Aid (FDA) OneWorld GUI development tool for building interactive applications and forms.

form interconnection Allows one form to access and pass data to another form. Can be attached to any event. Are normally used when a button is clicked.

form type The following form types are available in OneWorld:

- 1) Find/browse
- 2) Fix/inspect
- 3) Header detail
- 4) Headerless detail
- 5) Message
- 6) Parent/child
- 7) Search/select

Fourth Generation Language (4GL) Programming language that focuses on what you need to do and determines how to do it. Structured Query Language is an example of a 4GL.

FTP File Transfer Protocol

GBR General Business Rule. See *Event Rules*.

Global Build The building of all business function objects and DLLs.

Global Table Specs The specifications that OneWorld needs to access a database table. OneWorld builds these from a variety of other table specifications as

needed. The first time a table is accessed, OneWorld builds the global table specs for that table.

graphical user interface (GUI) A computer interface that is graphically-based as opposed to being character-based. An example of a character-based interface is that of the AS/400 system. An example of a GUI is Microsoft Windows. Graphically-based interfaces allow pictures and other graphic images to be used to give people clues on how to operate the computer.

GUI See *graphical user interface*.

HA High Availability

HLMI High Level Machine Interface. See *Technology Independent Machine Interface*.

host In the centralized computer model, a large timesharing computer system that terminals communicate with and rely on for processing. Contrasts with client/server in that those users work at computers that perform much of their own processing and access servers that provide services such as file management, security, and printer management.

HR Human Resources

HTML See *HyperText Markup Language*.

HyperText Markup Language (HTML) A markup language used to specify the logical structure of a document rather than the physical layout. Specifying logical structure makes any HTML document platform independent. You can view an HTML document on any desktop capable of supporting a browser. HTML can include active links to other HTML documents anywhere on the Internet or on intranet sites.

IBM International Business Machines Corporation

IFS See *integrated file system*.

IGS IBM Global Services

ILE Integrated Language Environment

I-listed IBM-listed

index Represents both an ordering of values and a uniqueness of values that provide efficient access to data in rows of a table. Is made up of one or more columns in the table.

integrated file system The overarching file system that provides a common interface to all of the subsidiary file systems on the AS/400 system, including "root" (DOS-like), QOpenSys (UNIX-like), QSYS.LIB (the traditional AS/400 library-based file system), and several others.

integrated toolset An industrial-strength toolset (unique to OneWorld) embedded in the already comprehensive business applications. Is the same toolset used by J.D. Edwards to build OneWorld interactive and batch applications. Much more than a development environment, the OneWorld integrated toolset handles reporting and other batch processes,

change management, and basic data warehousing facilities.

interactive processing Processing actions that occur in response to commands you enter directly into the system. During interactive processing, you are in direct communication with the system, and it may prompt you for additional information while processing your request. Contrasts with batch processing. See also *online*.

interface A link between two or more computer systems that allows these systems to send information to and receive information from one another.

Internet Worldwide constellation of servers, applications, and information available to a desktop client through a phone line or other type of remote access.

interoperability Ability of different computer systems, networks, operating systems, and applications to work together and share information.

intranet Small version of the Internet usually confined to one company or organization. An intranet uses the functionality of the Internet and places it at the disposal of a single enterprise.

IOP Input/Output Processor

IP Connectionless communication protocol that, by itself, provides a datagram service. Datagrams are self-contained packets of information that are forwarded by routers based on their address and the routing table information contained in the routers. Every node on a TCP/IP network requires an address that identifies both a network and a local host or node on the network. In most cases, the network administrator sets up these addresses when installing new workstations. In some cases, however, it is possible for a workstation, when booting up, to query a server for a dynamically assigned address.

IPCS Integrated PC Server

IPDS Intelligent Printer Data Stream

IPX Internetwork Packet eXchange

ISDN Integrated Services Digital Network

IServer Service Internet server service (developed by J.D. Edwards) that resides on the Web server. Is used to speed up delivery of the Java class files from the database to the client.

ISV Independent Software Vendor

ITSO International Technical Support Organization

J.D. Edwards Database See *JDEBase Database Middleware*.

Java Internet executable language that, like C, is designed to be highly portable across platforms. This programming language was developed by Sun Microsystems. Applets, or Java applications, can be accessed from a Web browser and executed at the

client, provided that the operating system or browser is Java-enabled (Java is often described as a scaled-down C++). Java applications are platform independent.

Java Database Connectivity (JDBC) The standard way to access Java databases, as set by Sun Microsystems. Allows you to use any JDBC driver database.

JDBC Java Database Connectivity

JDBNet The OneWorld database driver that allows heterogeneous servers to access each other's data. For example, an AS/400 can access Oracle data on an HP9000 using JDBNet. JDBNet uses JDENet to communicate with other servers.

JDE.INI J.D. Edwards file (or member for the AS/400 system) that provides the runtime settings required for OneWorld initialization. Specific versions of the file/member must reside on every machine running OneWorld. This includes workstations and servers.

JDEBase Database Middleware J.D. Edwards proprietary database middleware package that provides two primary benefits:

- Platform-independent APIs for multidatabase access. These APIs are used in two ways:
 - By the interactive and batch engines to dynamically generate platform-specific SQL, depending on the data source request.
 - As open APIs for advanced C business function writing. These APIs are then used by the engines to dynamically generate platform-specific SQL.
- Client-to-server and server-to-server database access. To accomplish this, OneWorld is integrated with a variety of third-party database drivers, such as Client Access/400 and open database connectivity (ODBC).

JDEIPC This is the interprocess communication APIs within OneWorld. These functions are platform independent and allow processes to communicate with each other through message queues, shared memory, semaphores, etc.

JDENet communications middleware J.D. Edwards proprietary communications middleware package for OneWorld. A peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all OneWorld supported platforms.

JFC Java Foundation Classes

JITI See *Just-In-Time Installation*

JITR See *Just-In-Time Replication*

job queue A group of jobs waiting to be batch processed. See also *batch processing*.

Just-In-Time Installation (JITI) OneWorld's method of dynamically replicating objects from the central object location to a workstation as needed. For example, if a user has installed a partial package and wants to run an application that was not delivered in the package, OneWorld will automatically install what it needs to run the application.

Just-In-Time Replication (JITR) OneWorld's method of replicating data to individual workstations. OneWorld replicates new records (inserts) only at the time the user needs the data. Changes, deletes, and updates must be replicated using Pull Replication.

JVM Java Virtual Machine

kernel Not an OS kernel, rather the OneWorld foundation code. This is where the database middleware, the caching code, the business function calling mechanism, and other lower level functions are located.

kernel job One of two types of jdenet processes. Kernel jobs are the processes on an Enterprise Server that really do the work of calling a business function, submitting a UBE, retrieving data, etc. There can be 0 to many kernel jobs running on an Enterprise Server. The actual executable is jdenet_k.

kernel type On a OneWorld Application Server, kernel jobs of different types can be started. For example, Security Server uses a different kernel type than JDBNET does.

KEY A column or combination of columns that identify one or more records in a database table.

LAN Local area network

LAPD Link Access Procedure for D-channel of ISDN

LF Logical File

LIC Licensed Internal Code

logic server This term is often used as a synonym for Application Server or Enterprise Server. This phrase would never be used to describe a data-only server.

LPD Line Printer Daemon

LPO Licensed Program Offering

LPQ Line Printer Queue

LPR Line Printer Requester

LUW Logical Unit of Work

Management Central A suite of system management functions known as Management Central has been integrated into Operations Navigator.

MAPI See *Messaging Application Programming Interface*.

Master Business Function (MBF) A piece of business logic that can be mapped to run either on the workstation or on an Application Server. The mapping is accomplished through the Object Configuration Manager (OCM).

master table A database table used to store data and information that is permanent and necessary to the system's operation. Master tables may contain such data as paid tax amounts, supplier names, addresses, employee information, and job information.

MBF See *Master Business Function*.

MBF menu A menu that displays numbered selections. Each of these selections represents a program or another menu. To access a selection from a menu, type the selection number and then press Enter.

menu levels See *level of detail*.

menu masking A security feature of J.D. Edwards systems that lets you prevent individual users from accessing specified menus or menu selections. The system does not display the menus or menu selections to unauthorized users.

Messaging Application Programming Interface (MAPI) An architecture that defines the components of a messaging system and how they behave. Defines the interface between the messaging system and the components.

MI Machine Interface

middleware A general term that covers all distributed software needed to support interactions between clients and servers. Think of it as the software that is in the middle of the client/server system or the "glue" that lets the client obtain a service from a server.

MTU Maximum Transmission Unit

multi-tier architecture Client/server architecture that allows multiple levels of processing. A tier defines the number of computers that can be used to complete some defined task.

national language support (NLS) OneWorld supports many national languages and cultural characteristics with a common set of program code. For the user interface, different sets of textual data is used for different languages. For character data sent to and received from different platforms, OneWorld performs the necessary data conversion to preserve data integrity.

net job One of two types of jdenet processes. Net jobs are the processes on the server that handle all of the network communications. There can be one to many net jobs running on an Enterprise Server. The actual executable is jdenet_n.

network computer As opposed to the personal computer, the network computer offers (in theory) lower cost of purchase and ownership and less complexity. Basically, it is a scaled-down PC (very little memory or disk space) that can be used to access network-based applications (Java applets, ActiveX controls) through a network browser.

network computing Often referred to as the next phase of computing after client/server. While its exact definition remains obscure, it generally encompasses issues such as transparent access to computing resources, browser-style front-ends, platform independence, and other similar concepts.

Next numbers A feature you use to control the automatic numbering of such items as new G/L accounts, vouchers, and addresses. It lets you specify a numbering system and provides a method to increment numbers to reduce transposition and typing errors.

NFS Network File System

NIC Network Interface Card

NLS See *national language support*.

NPT Non-programmable Terminal

object A self-sufficient entity that contains data, as well as the structures and functions used to manipulate the data. For OneWorld purposes, an object is a reusable entity that is based on software specifications created by the OneWorld toolset. See also *object librarian*.

Object Configuration Manager (OCM) OneWorld's Object Request Broker and the control center for the runtime environment. Keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, the Object Configuration Manager directs access to it using defaults and overrides for a given environment and user.

object embedding When an object is embedded in another document, an association is maintained between the object and the application that created it. However, any changes made to the object are also only kept in the compound document. See also *object linking*.

Object librarian A repository of all versions, applications, and business functions reusable in building applications. Provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of OneWorld Objects. Supports multiple environments (such as production and development) and allows objects to be easily moved from one environment to another.

object linking When an object is linked to another document, a reference is created with the file the object is stored in, as well as with the application that created it. When the object is modified, either from the compound document or directly through the file it is saved in, the change is reflected in that application as well as anywhere it has been linked. See also *object embedding*.

object linking and embedding (OLE) A way to integrate objects from diverse applications, such as graphics, charts, spreadsheets, text, or an audio clip

from a sound program. See also *object embedding* and *object linking*.

object-based technology (OBT) Technology that supports some of the main principles of object-oriented technology: classes, polymorphism, inheritance, or encapsulation.

object-oriented technology (OOT) Brings software development past procedural programming into a world of reusable programming that simplifies development of applications. Object orientation is based on the following principles: classes, polymorphism, inheritance, and encapsulation.

OCI Oracle Call Level Interface

OCM See *object configuration manager*.

ODBC See *open database connectivity*.

OLAP On-line Analytical Processing

OLE See *object linking and embedding*.

OLTP On-line Transaction Processing

OneWorld A combined suite of comprehensive, mission-critical business applications and an embedded toolset for configuring those applications to unique business and technology requirements. Is built on the Configurable Network Computing technology, J.D. Edwards' own application architecture, which extends client/server functionality to new levels of configurability, adaptability, and stability.

OneWorld application Interactive or batch processes that execute the business functionality of OneWorld. Consist of reusable business functions and associated data that are platform independent and can be dynamically configured across a TCP/IP network.

OneWorld object A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, Data Dictionary items, batch processes, business views, event rules, versions, data structures, and media objects. See also *object*.

OneWorld process Allows OneWorld clients and servers to handle processing requests and execute transactions. A client runs one process, and servers can have multiple instances. OneWorld processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes do not have to wait if the server is particularly busy.

OneWorld Tools The OneWorld CASE tools. See also *BDA*, *FDA*, *RDA*, and *TDA*.

OneWorld Web development computer A standard OneWorld Windows developer computer with the additional components installed:

- JFC (0.5.1)
- Generator Package with Generator.Java and JDECOM.dll

- R2 with interpretive and application controls/form.

online Computer functions over which the system has continuous control. Users are online with the system when working with J.D. Edwards system provided forms.

open database connectivity (ODBC) Defines a standard interface for different technologies to process data between applications and different data sources. Interface is made up of a set of function calls, methods of connectivity, and representation of data types that define access to data sources.

Open Systems Interconnection (OSI) A model developed by the International Standards Organization (ISO) in the early 1980s. It defines protocols and standards for the interconnection of computers and network equipment.

Operations Navigator AS/400 Operations Navigator is the strategic platform for providing a graphical interface to systems administration functions and is a feature of AS/400 Client Access Express for Windows.

OPM Original Program Model

OptiConnect OptiConnect is a combination of hardware and software (available only on the AS/400) that allows you to connect multiple high-end AS/400 systems by providing high-speed transparent access to data through fiber optic bus connections. OptiConnect is also the term used to refer to the fee-based part of OS/400 that supports the OptiConnect hardware.

OptiMover OptiMover is an IBM PRPQ which is the subset of OptiConnect software that supports TCP/IP. OptiMover software is the minimum software required to run OneWorld over while the full OptiConnect software (TCP/IP and DDM) is required for World or coexistence environments.

OS/400 Operating System/400

OSC OptiConnect Services Center

OSI Open Systems Interconnection

output Information that the computer transfers from internal storage to an external device, such as a printer or a computer form.

output queue See *print queue*.

package OneWorld objects are installed to workstations in packages from the Deployment Server. Can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the Deployment Server the install program can find them. It is a point-in-time "snap shot" of the central objects on the Deployment Server.

package build The process, in OneWorld, of gathering together all of the components that will later be deployed to a server or client.

package location The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path_code\package\package_name. The sub-directories under this path are where the replicated objects for the package will be placed. Also referred to as where the package is built or stored.

Paper Fix Instructions for modifying an application to fix a bug. Generally, these are sent to customers by the application development groups. The tools group will usually have to send an external service pack to customers.

partitioning Technique for distributing data to local and remote sites to place data closer to the users who access it. Portions of data can be copied to different database management systems.

Path code Pointer to a specific set of objects. Used to locate: Central Objects and Replicated Objects.

platform independence A benefit of open systems and Configurable Network Computing. Applications that are composed of a single code base can be run across a TCP/IP network consisting of various server platforms and SQL databases.

PCF Product Configurator

PCL Printer Control Language

PCML Program Call Markup Language

PDML Panel Definition Markup Language

PEX Performance Explorer

PF Physical File

PID Process Identifier

PING Packet InterNet Groper

PMR Problem Management Record

POP Post Office Protocol

portability Allows the same application to run on different operating systems and hardware platforms.

POSIX Portable Operating System Interface

PPP Point-to-Point Protocol

primary key A column or combination of columns that uniquely identifies each row in a table.

print queue A list of tables, such as reports, that you have submitted to be written to an output Configurable Network Computing Implementation device, such as a printer. The computer spools the tables until it writes them. After the computer writes the table, the system removes the table identifier from the list.

pristine The term used to describe an environment that is used as a point of reference and is never changed. This provides a place to retrieve copies of original objects in the case something is damaged or inadvertently changed.

processing option A feature of the J.D. Edwards reporting system that allows you to supply parameters to direct the functions of a program. For example, processing options allow you to specify defaults for certain form displays, control the format in which information prints on reports, change how a form displays information, and enter beginning dates.

program temporary fix (PTF) Representation of changes to J.D. Edwards software that your organization receives on magnetic tapes or diskettes.

proxy When a user with user ID X signs on and is served up a substitute user ID Y, or proxy, and user ID X really runs under the attributes and authority of Y. See *security profile* for a OneWorld example.

PRPQ Programming Request for Price Quotation

PSF Print Services Facility

PSP Preventive Service Planning

PTF Program Temporary Fix

published table Also called a “Master” table. This is the central copy to be replicated to other machines. Resides on the “Publisher” machine. The Data Replication Publisher Table (F98DRPUB) identifies all of the published tables and their associated Publishers in the enterprise.

publisher The server that is responsible for the Published Table. The Data Replication Publisher Table (F98DRPUB) identifies all of the Published Tables and their associated Publishers in the enterprise.

pull replication One of the OneWorld methods for replicating data to individual workstations. Such machines are set up as Pull Subscribers using OneWorld’s data replication tools. The only time Pull Subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the Pull Subscriber to the server machine that stores the Data Replication Pending Change Notification table (F98DRPCN).

QBE See *Query by Example*.

QM Queuing Multiplier

QSH Qshell

Query by Example (QBE) Located at the top of a detail area. Is used to search for data to be displayed in the detail area.

RAID Redundant Array of Independent Disks

RDA See *Report Design Aid*.

RDB Tables This does not always refer to all tables in a relational database within the contexts of OneWorld. Rather, it is referring to the central repository for the TAM files. All TAM files are stored centrally in relational database files. TAM is actually built from these RDB tables in the package build process.

RDBMS Relational Database Management System

redundancy Storing exact copies of data in multiple databases.

regenerable Source code for OneWorld business functions can be regenerated from specifications (business function names). Regeneration occurs whenever an application is recompiled, either for a new platform or when new functionality is added.

relationship Links tables together and facilitates joining business views for use in an application or report. Relationships are created based on indexes.

release/release update A “release” contains major new functionality, and a “release update” contains an accumulation of fixes and performance enhancements, but no new functionality.

replicated object A copy or replicated set of the central objects must reside on each client and server that run OneWorld. The path code indicates the directory the directory where these objects are located.

Report Design Aid (RDA) This is the OneWorld CASE tool used to design batch processes or reports (UBEs). See also *BDA*, *FDA*, and *TDA*.

RFS Remote File System

RIP Routing Information Protocol

RISC Reduced Instruction Set Computing

RPC Remote Procedure Call

run To cause the computer system to perform a routine, process a batch of transactions, or carry out computer program instructions.

Runtime The application engine. This is the piece of the system code which drives the applications via event rules that are stored in TAM.

SAR See *Software Action Request*.

SAW See *Server Administration Workbench*.

SBCS See *Single Byte Character Set*.

scalability Allows software, architecture, network, or hardware growth that will support software as it grows in size or resource requirements. The ability to reach higher levels of performance by adding processors.

SCS SNA Character String

SD Solution Developers

SDK Software Developers Kit

SDLC Synchronous Data Link Control

search/select A type of form used to search for a value and return it to the calling field.

security profile The security profile, or AS/400 system profile, is the profile under which processes are run for a OneWorld user. OneWorld stores the security profile in the security server file F98OWSEC. There can be one or many security profiles. The

security profile is only used if the security server kernel is running.

Security Server The OneWorld server performing sign-on security and database access security. If the security server is not up and running nobody can get into OneWorld.

Security Workbench The OneWorld application for security administrators to control authority for individual users and groups to various features such as applications, table rows, processing options and so on.

selection Found on J.D. Edwards menus. Represent functions that you can access from a menu. To make a selection, type the associated number in the Selection field, and press Enter.

server Provides the essential functions for furnishing services to network users (or clients) and provides management functions for network administrators. Some of these functions are:

- Storage of operating system program modules, utilities, and commands
- Storage of user programs and data
- Management functions for the file system
- Management functions for security and user access
- Network monitoring and management components
- Data protection functions for fault tolerance

It may not be possible for one server to support all users with the required services. Dedicated servers that handle specific tasks, such as those listed here, are also common:

- Backup and archive servers
- Application Server
- Database server
- Fax server
- Print Server
- Electronic mail server
- Directory services server

Server Administration Workbench (SAW) OneWorld tool designed to make OneWorld server administration easier.

Server Package Install The mechanism used to get TAM and business function updates from Object Librarian to Enterprise Servers.

Server Map The OCM and data source tables for an Enterprise Server.

servlet Provides a Java-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions. An object that conforms to a specific interface that can be plugged into a

Java-based server. Is to the server-side as applets are to the client-side.

SETOWAUT An additional set of security tools that offers a comprehensive approach to securing OneWorld's user profiles and objects. This security model allows OneWorld users to have access to OneWorld objects while excluding *PUBLIC access.

SEU Source Entry Utility

Single Byte Character Set (SBCS) Text character sets in which each character can be identified with a single byte.

SLIC See *System Licensed Internal Code*.

SLS Single Level Storage

SMAPP System Managed Access Path Protection

SMB System Message Block

SMP Symmetric Multiprocessing Processor

SMTP Simple Mail Transfer Protocol

SNA Systems Network Architecture

SOI Silicon on Insulator

Snapshot The process which stores an existing OneWorld installation on a client PC so that a user can run another installation. Snapshot allows the user to quickly and easily switch between versions of OneWorld.

SNMP Simple Network Management Protocol

Software Action Request (SAR) An entry in the AS/400 database used for requesting modifications to J.D. Edwards software.

software Operating system and application programs that tell the computer how and what tasks to perform.

Software Protection Codes (SPC) These are authorization codes which are set up at a customer site as part of the initial install of OneWorld. Without correct codes, the customer will not be able to install OneWorld.

SPC See *Software Protection Codes*

Spec Files See *Table Access Management*

Spec Install Transferring the TAM specifications for a version of a report from a client PC to an Enterprise Server when that version is being run on a server.

specifications Complete description of a OneWorld object. Each object has its own specification, or name, which is used to build applications.

spool The function by which the system stores generated output to await printing and processing.

spooled table A holding file for output data waiting to be printed or input data waiting to be processed.

SPX Sequenced Packet eXchange

SQL See *Structured Query Language*.

SSL Secure Sockets Layer

SST System Service Tools

static text Short, descriptive text that appears next to a control variable or field. When the variable or field is enabled, the static text is black. When the variable or field is disabled, the static text is gray.

Structured Query Language (SQL) A fourth generation language used as an industry standard for RDBA. Can be used to create databases and to retrieve, add, modify, or delete data from databases. Is not a complete programming language because it does not contain control flow logic.

subfile See *detail*.

submit See *run*.

subscriber The server that is responsible for the replicated copy of a Published Table. Such servers are identified in the Subscriber Table.

subscriber table The subscriber table (F98DRSUB), which is stored on the Publisher Server with the Data Replication Publisher Table (F98DRPUB) identifies all of the subscriber machines for each published table.

summary Presentation of data or information in a cumulative or totaled manner in which most of the details have been removed. Many of the J.D. Edwards systems offer forms and reports that are summaries of the information stored in certain tables. Contrasts with *detail*.

system See *application*.

system code A numerical representation of J.D. Edwards and customer systems. For example, 01 is the system code for Address Book. System codes 55 through 59 are reserved for customer development by customers. Use system codes to categorize within OneWorld. For example, when establishing user defined codes (UDCs), you must include the system code the best categorizes it. When naming objects such as applications, tables, and menus, the second and third characters in the object's name is the system code for that object. For example, G04 is the main menu for Accounts Payable, and 04 is its system code.

system function A program module, provided by OneWorld, available to applications and reports for further processing.

System Licensed Internal Code (SLIC) Internal code running on AS/400 RISC systems.

table A two-dimensional entity made up of rows and columns. All physical data in a database are stored in tables. A row in a table contains a record of related information. An example is a record in an Employee table containing the Name, Address, Phone Number, Age, and Salary of an employee. Name is an example of a column in the employee table.

Table Access Management (TAM) The OneWorld component that handles the storage and retrieval of

user defined specifications. TAM stores information such as Data Dictionary definitions, application and report specifications, event rules, table definitions, business function input parameters and library information, and data structure definitions for running applications, reports, and business functions.

Table Conversion (TC) A OneWorld mechanism to convert non-OneWorld or "foreign" tables to OneWorld and vice-versa.

Table Design Aid (TDA) OneWorld GUI tool for creating, modifying, copying, and printing database tables.

table event rules Are used to attach database triggers (or programs) that automatically run whenever an action occurs against the table. An action against a table is referred to as an event. When you create a OneWorld database trigger, you must first determine which event will activate the trigger. Then, use Event Rules Design to create the trigger. Although OneWorld allows event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.

TAM See *Table Access Management*.

TC See *Table Conversion*.

TCP/IP See *Transmission Control Protocol/Internet Protocol*.

TCP/IP services port Used by a particular server application to provide whatever service the server is designed to provide. The port number must be readily known so that an application programmer can request it by name.

TDA See *table design aid*.

TDQ Task Dispatching Queue

Technology Independent Machine Interface (TIMI) A software-based interface that shields the operating system and application programs from hardware technology changes on the AS/400 system.

TELNET Teletypewriter Network

terminal identification The workstation ID number. Terminal number of a specific terminal or IBM user ID of a particular person for whom this is a valid profile. *Header Field:* Use the Skip to Terminal/User ID field in the upper portion of the form as an inquiry field in which you can enter the number of a terminal or the IBM user ID of a specific person whose profile you want the system to display at the top of the list. When you first access this form, the system automatically enters the user ID of the person signed on to the system. *Detail Field:* The Terminal/User ID field in the lower portion of the form contains the user ID of the person whose profile appears on the same line. A code identifying the user or terminal for which you accessed this window.

TFTP Trivial File Transfer Protocol

thin client A client that is used primarily as an intelligent graphical display device, while most, if not all, application logic is executed on the server.

Third Generation Language (3GL) Programming language that requires detailed information about how to complete a task. Examples of 3GLs are COBOL, C, Pascal and FORTRAN.

TIMI See *Technology Independent Machine Interface*.

Transaction Processing A term used to describe the process of defining the beginning of a group of database updates and their eventual commitment or rollback. The purpose of using transaction processing is to ensure integrity of the database at the application document level.

Transmission Control Protocol/Internet Protocol (TCP/IP) The original TCP protocol was developed as a way to interconnect networks using many different types of transmission methods. TCP provides a way to establish a connection between end systems for the reliable delivery of messages and data.

trigger Allows you to attach default processing to a data item in the Data Dictionary. When that data item is used on an application or report, the trigger is invoked by an event associated with the data item. OneWorld also has three visual assist triggers: calculator, calendar, and search form.

TSC Technology Solutions Center

TSE Terminal Server Edition

UBE See *Universal Batch Engine*.

UDB Universal Database

UDC See *User Defined Codes*.

UDC Edit Control Is used for a field that accepts only specific values defined in a UDC table. Associate a UDC edit control with a database item or dictionary item. The visual assist Flashlight automatically appears adjacent to the UDC edit control field. When you click on the visual assist Flashlight, the attached search and select form displays valid values for the field. To create a UDC Edit Control, you must associate the data item with a specific UDC table in the Data Dictionary. Then, create a search and select form for displaying valid values from the UDC table.

UDFS User-defined File System

uniform resource locator (URL) Names the address of a document on the Internet or an intranet.

Universal Batch Engine (UBE) OneWorld Software that prints reports and does batch processing. Work done by UBE for a specific run is based upon a set of specifications entered through Report Design Aid.

UPS Uninterruptible Power Supply

URL See *Uniform Resource Locator*.

user defined code (type) Identifier for a table of codes with a meaning you define for the system, such

as ST for the Search Type codes table in Address Book. J.D. Edwards systems provide a number of these tables and allow you to create and define tables of your own. Were formerly known as “descriptive titles”.

user defined codes (UDC) Codes within software that users can define, relate to code descriptions, and assign valid values. Sometimes user defined codes are referred to as a generic code table. Examples of such codes are unit-of-measure codes, state names, and employee type codes.

valid codes The allowed codes, amounts, or types of data that you can enter in a field. The system verifies the information you enter against the list of valid codes.

visual assist Forms that can be invoked from a control to assist the user in determining what data belongs in the control.

vocabulary overrides Feature you can use to override field, row, or column title text on forms and reports.

WAN Wide area network

Web client Any workstation that contains an Internet browser. Communicates with the Web server for OneWorld data.

Web server Any workstation that contains the IServer service, SQL server, Java menus and applications, and Internet middleware. Receives data from the Web client, and passes the request to the Enterprise Server. When the Enterprise Server processes the information, it sends it back to the Web server, and the Web server sends it back to the Web client.

window See *form*.

workflow According to the Workflow Management Coalition, workflow means “the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.”

workgroup server A remote Database Server usually containing subsets of data replicated from a master Database Server. This server does not perform application or batch processing. It may have OneWorld running (to replicate data).

World Wide Web Part of the Internet that can transmit text, graphics, audio, and video. The World Wide Web allows clients to launch local or remote applications.

WSG Workstation Gateway

WTS Windows Terminal Server

XMA Extended Middleware Architecture

XML Extensible Markup Language

XPG X/OPEN Portability Guide

Index

Symbols

*ADD 418
*AFPDS 383
*AFPDSLIN 383
*ALLIO 92
*ALLOBJ 416, 417
*AUDIT 416
*AUTLMGT 418
*BASE 110
*DELETE 419
*EXECUTE 419
*FILE object 627
*FIRSTIO 92
*HPPJLDRV 381
*IBMPJLDRV 381
*IOSYSCFG 416, 782
*IPDS 383
*JOBCTL 416
*JRN 461
*JRNRCV 461
*LINE 383
*MACHINE 110
*OBJALTER 418
*OBJEXIST 418
*OBJMGT 418
*OBJOPR 418
*OBJREF 418
*PGMR 416
*PUBLIC 419
*READ 418
*SAVSYS 416
*SCS 383
*SECADM 416
*SECOFR 416
*SERVICE 416
*SHRPOOL1 110
*SHRPOOL60 110
*SPLCTL 416
*SQL 88
*SYMLNK 39
*SYS 88
*SYSOPR 416
*UPDATE 418
*USER 416
*USERASCII 384
[SECURITY] section 324, 429

Numerics

5769-ST1 86
64-bit 13

A

A98OWMNU 299
access path 84, 99, 469
access path sharing 589
access path size 588

Action Security 429
Active status 118
Active-ineligible status 119
ActivEra 3
Active-wait status 119
activity level 116
Activity levels 118
Add RDB Directory Entry 188
ADDPEXDFN command 652
ADDRDBDIRE command 188
address 11
adopted authority 419
ADSM 461
ADSTAR Distributed Storage Manager 461
Advanced Function Printing 23
Advanced Function Printing Data Stream 383
Advanced Technical Support 826
Advised Index 646
AFP 23
AFPDS line data stream 383
Alternate IPL device 469
analytical applications 843
Analyze Results 644
analyzing SQL Performance Monitor results 644
APAR 139, 833, 834
API 116, 515
APIs
 QSQCHKS 87
 QSQPRCED 87
 XDA family of APIs 87
application data 56
application development process 331
Application Development Toolset 465
application installation 64
Application Programming Interface 116
Application Security 429
Application Server 54, 58, 71
application software 462
applications interoperability 843
APYJRNCHG 469
archiving solution 18
AS/400
 architecture 9
 client/server capability 8
 clustering 22
 data management 15
 database 14
 DB2 UDB for AS/400 14
 I/O architecture 10
 integrated file system 17
 integration 7
 interoperability 7
 job management 22
 Logical Partitioning 18
 machine interface 9
 network attributes 109
 networks 7
 object-based architecture 11
 printer management 23
 security 16

- SQL standards 15
- subsystems 110
- system management 22
- system values 109
- Technical Support Web site 834
- user management 23
- user profiles 16
- AS/400 data sources 62
- AS/400 Enterprise Server 141, 143
- AS/400 INI file settings 734
- AS/400 Institute 833
- AS/400 printing process 379
- AS/400 server tuning
 - assigning storage pool 576
 - expert cache 578
 - storage pool 576
- AS/400 shutdown 308
- AS/400 Support Web sites 833
- AS/400 system values 222
- AS/400 Technology Solutions Center 824
- AS/400 Web site 829
- ASCII data stream 384
- ASCII printers 381
- ASP 13, 766, 775
- assembling a package 344
- asynchronous I/O 612
- ATS 826
- attended IPL 775
- Audit Journal 419
- authorities 417
- authority management 16
- authorization 17
- authorization list 419
- Authorized Problem Analysis Report 139
- Authorized Program Analysis Report 834
- auto-commit 105
- Autostart 117
- autostart batch job 117
- autostart job 111, 115
- auxiliary storage pool 13, 773, 775
- advanced planning and scheduling 843

B

- B7332SYS 42
- backup 23
- backup and recovery 519
 - minimizing windows 20
- Backup and Recovery Media Services/400 455
- Backup Recovery and Media Services 461
- base pool 110
- batch immediate job 117
- batch job 117
- batch processes 321
- batch processing 72
- Batch Versions menu 408
- batch-of-one processing, asynchronous processing 50
- BCP 540
- BI 27
- BI (Business Intelligence) 27
- BM Technical support phone numbers 829

- BRMS 461
- BRMS/400 455
- BSFN 762
- Bulk Copy Program 540
- Business Data 59
- Business Intelligence (BI) 27
- business processing 72
- byte-string 11

C

- CACI 663
- CASE 3
- catalog table 90
- catch-up mode 97
- Cause 741
- CBT 837
- CCSID 215, 817
- CCSID (coded character set identifier) 635
- Centers for IBM e-business Innovation 827
- central object 63
- Central Object Specifications by Release 61
- central objects 34
- central server 54
- CFGTCP 785, 787, 815
- Change Query Attributes (CHGQRYA) command 590
- changing query attributes 590
- check in process 34
- check out process 34
- CHGJOB 753
- CHGJOB 730
- CHGPERIOD 468
- CHGQRYA 625
- CHGQRYA (Change Query Attributes) command 590
- CHGSECAUD command 419
- CHGUSRPRF command 417
- CHRID 817
- Citrix 522
- class 112, 118
- Cleaning up the OneWorld environment 320
- Clear IPC 320
- Client Access Express 522
 - benefits 290
 - connectivity 291
 - Data Transfer 291
 - license information 296
 - plug ins 285
 - transition from Client Access/400 292
- Client Access home page 746
- Client Access service pack 746
- Client Access/400 290
- client workstation 142, 143
- client/server performance analysis
 - data collection 605
 - performance data 606
 - performance tools/400 606
- client/server performance tuning
 - AS/400 server 567
- CLRIPC 233, 306, 321
- CLRPOOL command 591
- CMDWTR (Command Writer) command 393

- CNC (Configurable Network Computing) 47
- coded character set identifier (CCSID) 635
- coexistence 101, 247
 - planning 145
- cohabitation 247
- collection 87
- column 88
- Column Security 429
- columns 83, 84
- Command Writer (CMDWTR) command 393
- command, CL
 - Change Query Attributes (CHGQRYA) 590
 - CHGQRYA (Change Query Attributes) 590
 - Print Structure Query Language Information (PRTSQLINF) 621
 - Print Transaction Report (PRTTNSRPT) 613
 - PRTSQLINF (Print Structure Query Language Information) 621
 - PRTTNSRPT (Print Transaction Report) 613
- Commercial Processing Workload (CPW) 569
- commit 105, 633
- commit group 634
- commit mode 633
- commitment control 102, 633
- COMMON 834
- communication analysis tool 660
- communications entries 111
- communications trace 746
- compile 16
- Completion 740
- Configurable Network Computing (CNC) 47
- configure TCP/IP 781
- console 113
- consolidating worldwide operations 21
- Consult Line 832
- control data 72
- Control Table Workbench 488
- controlling subsystem 113
- copper chip technology 13
- Copy from Import File command 539
- Copy to Import File command 539
- cost center security 429
- CPW (Commercial Processing Workload) 569
- CPYFRMIMPF 539
- CPYTBL 537
- CPYTOIMPF 539
- CREATE TABLE 627
- CREATE VIEW 627, 629
- CRTLF 627, 629
- CRTLINETH 783
- CRTOUTQ 382
- CRTOWADPRF 452
- CRTPF 627
- CRTSQLPKG 88
- CRTUSRPRF command 417
- cultural job attribute values 473
- cultural system values 472
- cumulative update package 328
- Custom Documentation Tool 837
- Custom modifications 335

- Custom Plan 189
- customer preparation 140
- CWBODCB.DLL 295

D

- DASD 11
- data collection 843
- data definition language (DDL) 89
- Data Description Specifications 84
- Data Dictionary 433
 - by release 59
- data management processing 72
- data manipulation language (DML) 89
- data mining 28
- data replication 28, 50, 471, 529
- Data Replication Change Log 62
- Data Replication Pending Change Notifications 62
- data source 57, 101
 - definition 57
 - planning 146
 - translation 635
- Data Source Master 62
- data stream 383
- data warehousing 49, 844
- database 83, 462
 - catalog views 90
 - file 15, 84
 - indexes 588
 - J.D. Edwards naming conventions 98
 - monitor 766
 - query optimization 91
 - server 71
- database administrator 73
- database connection jobs 726
- database data sources 57
- Database Server 71
- DATALINK 85
- datamarts 843
- DataMirror 533
- DataPropagator Relational 530
- DB2 for AS/400 14
- DB2 Multisystem 28, 87
- DB2 Query Manager for AS/400 86
- DB2 Symmetric Multiprocessing 87
- DB2 UDB for AS/400 83
- DB2/400 83, 84
- DBLOG 659
- DBMS 83
- DDL 89
- DDL (data definition language) 89
- DDM 84
- DDS 16, 84
- debug 752
- debug mode 636
- debug QZDASOINIT job 766
- debugging a business function 762
- decision support 844
- Dedicated Service Tools 775, 777, 778
- dedicated system 460, 463, 777
- default libraries 633

- default wait time 112
- defect 771, 831
- defining a package 342
- defining and building a package 341
- Deployment Server 71, 141, 142
- Deployment server 519
- Deployment Server Installation 152
- design tools 47
- development 522
- device file 84
- DFTRDBCOL 88
- Diagnostic 740
- diagnostic message 744
- disk fragmentation 13
- disk space 11
- disk space requirements 144
- disk space usage 766
- Display Active Prestart Job command 128
- display file 15
- Display IPC 320
- Display Stream File 736
- distributed data 72
- distributed data replication 844
- distributed processing 72
- DML 89
- DML (data manipulation language) 89
- DNT (Do Not Translate) 146
- Do Not Translate (DNT) 146
- domain name 782
- Domino clustering 21
- double-byte installation 509
- drives 520, 523
- DSPACTPJ 128
- DSPIPC 321
- DSPJOB 731
- DSPJOBLOG 732
- DSPLIND 771
- DSPMSG 738
- DSPNETA 771
- DSPSFWRSC 155
- DSPSTMF 736
- DST 775, 777, 778
- dynamic SQL 90

E

- e-business 26
- e-centers 827
- e-commerce 847
- ECS 23, 771, 831
- EDTRCYAP 470
- electronic commerce 844
- Electronic Customer Support 23, 831
- eligible for purge 112
- EMEA IBM/J.D. Edwards Pre-sales Support Center 824, 837
- Encoded Vector Index (EVI) 93
- encoding the passwords 324
- End Host Server command 123
- ENDCMNTRC 747
- ENDCMNTRC command 750

- ENDDBG 754
- ENDDBMON *ALL 608
- ENDHOSTSVR 123
- ENDNET 233
- ENDRMTSPT 772
- ENDSRVJOB 754
- Enterprise Resource Planning 13
- Enterprise Server 71
- environment 51
 - definition 51
 - revisions form 51
- Environment Checker 145
- environment tables 51
- environments 51, 521
- ERP 13
- Escape 740
- escape message 743
- EVI 93
- Exclusive Application Security 429
- Exit Security 429
- Expert Cache 24, 116, 576, 578, 590
- export management 845
- EXT value 741
- External Calls Security 430
- externally described file 16

F

- F0093 51
- F0094 51
- F00941 52
- F00942 52, 53
- F9650 339
- F98306 508
- F986101 52, 56, 62, 559
- F98611 57, 62, 557
- F986111 62
- F98DRLOG 62
- F98DRPCN 62
- Fast Path 432
- fat client 73
- fiber optic solutions 545
- field 88
- field level authority 629
- field naming convention 100
- fields 84
- file serving 387
- file type 84
- financial reporting 844
- fix 318
- foreign tables 538
- forms 845
- foundation code 47
- From Program 741
- full package 327

G

- gateway 797
- GH9011 421
- GH9013 398

- GH9052 303
- Global e-business Integration 827
- globalization 510
- GO BACKUP 459
- GO CMDRST 460
- GO CMDSAV 459
- GO LICPGM 154
- GO RESTORE 460
- GO SAVE 459
- GO TCPADM 787
- group job 117
- group profile 417, 421
- growth 707
- guideline 578

H

- hard link 39
- hardware requirement 140
- hazardous materials management 845
- health check tasks 309
- heavy client 73
- hierarchical directory structure 17
- Hierarchical Storage Management (HSM) 25
- high availability 547, 844
- history log 723
- homogeneity 49
- horizontal growth 547
- host installation process 232
- host name 782
- host print transform 384, 388, 413
- host table 785

I

- I/O architecture 10
- IBM
 - Product Support Services 840
 - Solution Assurance 839
 - Technical Sales Support 839
 - Web site 828
- IBM Direct 834
- IBM Education and Training Customer Service 834
- IBM Global Services 825
- IBM Net.Data 26
- IBM Web site 828
- IBM Worldwide Pre-Sales Technical Support 823
- IBM/ERP Pre-Sales Team 824
- IBM/J.D. Edwards International Competency Center 824, 837
- IBMLink 828
- ICC 824
- ICF 84
- ICF file 84
- Idea to Action 3
- IFS 17
- ILE 741
- imaging 845
- implementing an archiving solution 18
- import management 845
- index 99

- index melding 650
- indices 99
- ineligible 119
- Ineligible status 118
- Information 740
- Information Schema 90
- Informational APAR 717, 834
- Initial installation 64
- initial program load (IPL) 113, 393
- initial thread 116
- Inquiry 740
- installation 151, 730
 - process 151
- Installation Menu 299
- Installation Planner 185, 188
- Installation Workbench 184, 213
- instruction number 741
- INSWNTSVR command 157
- integrated file system 17, 736
- Integrated Netfinity Server 465, 513
 - checking status 324
 - operations 324
 - starting and stopping 325
- Integrated PC Server 513
- integration 7
- Intelligent Printer Data Stream 383
- interactive job 117
- International Competency Center 824
- International Technical Support Organization 826
- Interprocess Communication 821
 - structures 320
- IPC 821
 - locks 306
- IPC structures 320
- IPDS printers 380
- IPL (initial program load) 113, 393
- Istar 13
- ITSO 826

J

- J.D. Edwards
 - applications 4
 - education and training 837
 - support 836
 - Web site 837
- J.D. Edwards Customer Support 831
- J.D. Edwards education and training 837
- J.D. Edwards Support Structure 831
- J.D. Edwards Users Group 838
- Japan IBM/J.D. Edwards Competency Center 824
- Java 25
- Java Application Server 142, 144
- Java Virtual Machine (JVM) 25
- jde.ini 304
- jde.ini file 224, 226, 427
- jde.ini file setting 566
- jde.ini server file setting 585
- JDEBase 66, 68
- JDEDEBUG.LOG 566, 655
- JDEDST 230

- JDEFIN 230
- JDEHRM 230
- JDELINK 772
- JDEMFG 230
- JDENet 65, 66, 555, 586
- JDENET (*CLS) 43
- JDENET_K 133
- JDENET_N 133
- JDETECH 230
- JITI 65, 328
- JITI (Just-In-Time Installation) 328
- job 22, 730
 - description 118
 - log 730, 767
 - management 22
 - name 117
 - number 118, 733
 - priority 118
 - state 118
 - transitions 119
- Job Control Status Master 62
- Job ID 719
- job identification 719
- job name 733
- Job Number Master 62
- job queue 111, 230
- job trace 651, 751, 754
- job user 733
- journal 461, 469
- journal file analysis 667
- journal receiver 94, 461, 469, 773, 775
- journaling 94, 461
- Just-In-Time Installation (JITI) 328
- JVM (Java Virtual Machine) 25

K

- kernel process 68, 131
- keyed logical file 84

L

- lab exercise 646
- labor management 845
- Lakeview Technology 534
- Language ID 817
- language installation 484
- large objects (LOB) 85
- layered architecture 7
- LCKW 766
- library 87
- LIBRARY file 227
- Library List Control table 51
- Library List Master table 51
- LIC 10, 739
- Licensed Internal Code 9
- licensed programs 7
- limited capability (LMTCPB) 417
- line data stream 383
- line printer daemon (LPD) 386
- line printer requester (LPR) 386

- lite client 80
- LMTCPB 417
- localization 510
- lock manager 566
- locking 102
- log 767
- log files
 - location 736
 - viewing 736
- Log Manager 107
- logic data sources 58
- logic machine 58
- Logic Server 54, 58, 71
- logical file 84, 88, 99
- logical I/O 612
- Logical Partitioning (LPAR) 77, 547
- long wait 112
- LOOPBACK 790
- Lotus 838
- Lotus Domino 27
- LPAR (Logical Partitioning) 18, 77, 547
- LPD (line printer daemon) 386
- LPR (line printer requester) 386

M

- Machine Detail Master 339
- machine pool 110
- machine readable information (MRI) 471
- Management Central 24, 286
- manual commit 105
- manual IPL 775
- manufacturing execution system 846
- master business function 521
- maximum activity level 111
- maximum CPU time 112
- maximum field data returned 634
- maximum temporary storage 112
- memory 520, 523
- memory faulting rate 576
- memory management 116
- memory pools 116
- menu security 432
- message 312, 738, 739, 766
- message file 739
- Message ID 739
- message queue 738, 767, 821
- Message Type 739
- Metaframe 522
- Microsoft Access 183
- Microsoft Windows 8
- middleware 47, 65
- MIL3 663, 857
- minimizing backup and recovery windows 20
- mixed production environment 18
- modem 772
- modification rules 337
- Modify Selected Queries 645
- modless mods 337
- MRI (machine readable information) 471
- MSGW 766

- MTU size 566
- multidatabase applications 20
- multiple job queues 301
- multiple production environment 18
- multiple servers 50
- Multi-Tier Deployment 334
- multi-tiered networks 50

N

- naming convention 99
- national language support 23
- national language version 817
- net process 67, 130
- Net.Data 26
- NetServer 515
- network attributes 109
- network processes 67, 131
- network sniffer 662
- network-centric software 47
- NEXTHOP 815
- NLS 23
- Notify 740
- n-tier design 79

O

- object 11, 16
- Object Configuration Manager 51, 54, 62
- Object Configuration Master table 52
- object deployment 64
- Object Librarian by Release 61
- object locks 766
- Object Path Master application 53
- Object Path Master table 52
- object storage 63
- object-based architecture 11
- OCM 51, 54
- OCM mapping 241
- OCM table 559
- ODBC 130
 - components 293, 294
 - connection 580, 726
 - data source 295
 - driver 294
- ODBC connection job flow 295
- ODBC Data Source format parameters 634
- ODBC Data Source server parameters 632
- ODBC Data Sources (IBM provided) 631
- ODBC driver manager 294
- ODBC error message 745
- ODBC interface 293
- ODBC SQL package 621
- ODBC trace 654, 754
- ODBCLOG 659
- OMS/400 532
- OneWorld 4, 767
 - application development process 331
 - cleaning up the environment 320
 - data retrieval 56
 - design tools 47

- directories in the integrated file system 43
- foundation code 47
- implementation designs 71
- Installation Menu 299
- Interactive Runtime Engine 32
- libraries 39
- middleware 47, 65
- package management 327
- packages 43
- process flow 133
- system library 42
- OneWorld Client jde.ini file 669
- OneWorld Data Dictionary 254
- OneWorld environment 51
- OneWorld journal file analysis 667
- OneWorld table index analysis 668
- OneWorld upgrade process 246
- online job 726
- Open Data Path 623
- operating system 10
- Operations Console 285, 296
- Operations Navigator 284, 286, 292, 515, 733, 744
 - functions 286
 - tips and techniques 289
- operations schedule 301
- OPM (Original Program Model) 741
- Optimal Networks 662
- optimization process 625
- optimization strategy 92
- optimization tools 93
- optimized support 120
- OptiMover 546
- option 21 save 463
- Original Program Model (OPM) 741
- original support 120
- OS/400 739
- OS/400 communications trace 746
- OS/400 communications trace facility 660
- output queue 382

P

- P980042 53
- P98305 408
- P986110 55
- package 43, 88
 - assembling 344
- Package Assembly Director 343
- Package Build Definition Director 355
- Package Deployment Director 371
- Package Deployment Instructions 333
- package deployment methodology 332
- package management 327
- package types 327
- packages 43
- page faulting 578
- Parallel Data Loader 538
- parallel data loader 538
- Partial package 327
- partial package 327
- PartnerInfo 840

- PartnerWorld 827
- PartnerWorld for Developers 827
- PASS 389
- Passport Advantage 841
- password 23
- passwords encoding 324
- Path Code Master table 52
- path codes 52
- PC HelpCenter 826
- PDT 381, 388
- performance 8, 767
- performance collection files 636
- performance concepts 561
- performance data collection procedure 663
- Performance Explorer 652
- performance indicators 301
- Performance Tools/400 754
 - Component Report 611
 - Print Report 610
- PEX STATS 652
- PEX Trace 653
- PEXGUI 653
- physical file 84, 87, 88
- physical I/O 612
- PJ 724
- plan type 189
- Plan Validation 207
- planning check list 147
- PMR (Problem Management Record) 832
- pool size 111
- POP 787
- PORTTEST 233, 307, 755, 756
- presentation processing 72
- prestart 724
- prestart batch job 117
- prestart job 111, 125, 724
- prestart jobs 125, 305
- Preventive Service Planning (PSP) 718, 833
- print 23
- print management 846
- print serving 387
- Print Structure Query Language Information (PRTSQLINF)
 - command 621
- Print Transaction Report (PRTTNSRPT) command 613
- printed job log 733
- printer 23
- printer attachments 379
- Printer Definition Table 381, 388
- printer emulation 388
- printer file 15, 382
- Printer menu 397
- printer writer 385, 392
- printing 379
 - structure query language information 621
 - transaction report 613
- printing configuration 379
- printing rules 407
- private storage pool 111
- problem 771, 831
- problem determination checklists 711

- problem escalation contact list 300
- problem identification
 - aids 719
 - tips and techniques 760
- problem management 23
- Problem Management Record (PMR) 832
- problem prevention 717
- process flow 133
- processing mode 49
- Processing Option Security 429
- Processing Option Text table 508
- Product Support Services 840
- Program services 831
- program start request 111
- program temporary fix (PTF) 23, 831, 833
- PRTSQLINF 625
- PRTSQLINF (Print Structure Query Language Information)
 - command 621
- PRTTNSRPT (Print Transaction Report) command 613
- PSP (Preventive Service Planning) 718, 833
- PTF 23, 318, 831, 833
- Pulsar 13
- PWRDWNSYS 525, 777

Q

- QACTJOB 570
- QADLACTJ 570
- QADLTOTJ 570
- QAQQINI 92
- QAUDCTL 419
- QAUDLVL 419
- QAUTOCFG 382
- QAUTOCFG value 382
- QAUTORMT 382
- QAUTOVRT 382
- QBASACTLVL 570
- QBASE 113
- QBASPOOL 570
- QCMN 115
- QCPA 43
- QCPFMSG 739
- QCTL 113
- QCTLSBSD 113
- QDYNPTYSCD 570
- QESLINE 771
- QEZJOBLOG 733
- QINTER 114
- QJDEDST 304
- QJOBMSGQFL 570
- QJOBMSGQMX 570
- QMCHPOOL 570
- QNTC 515
- QPCSMPT 750
- QPFRADJ 570
- QPJOBLOG 733
- QPSRVTRC 754
- QQOOPTIONS 92
- QSECOFR 771, 782
- QSERVER 114, 125
- QSNADS 115

QSPL 114
 QSTRPRTWTR 393
 QSTRUPGM 113
 QSYSOPR 738, 766, 767
 QSYSWRK 114, 125
 QTCP 788
 QTCPIP 787
 QTOTJOB 570
 QTSEPOOL 570
 query attribute 624
 changing 590
 Query Optimizer 636
 query tools 846
 QUEST 838
 queuing concept 561
 queuing multiplier 563
 effect 562
 QUSRWRK 125
 QYPINT 591, 653, 857
 QZDAINIT/QZDASOINIT 576
 QZDASOINIT 305
 QZDASOINIT job 576, 726
 QZDASOINIT settings 580

R

RAID-5 568
 record 84, 88
 record change detection 103
 record locking 102
 recovery 741
 redbooks 827
 Reduced Instruction Set Computing (RISC) 10
 relational database 14
 Remote command 319
 remote journal 95
 remote output queue 389
 remote support 771
 remote system printing 385
 remote writer 386
 Replicated Local Data 59
 replicating TAM specifications 305
 Reply 740
 report printing 408
 report processing 72
 reporting tools 846
 Request 740
 request 561
 residencies 827
 resource name 783
 response time 766
 client/server response time 564
 restore 460
 restricted state 113, 460, 463
 RISC (Reduced Instruction Set Computing) 10
 RMVOWADPRF 453
 rollback 105, 633
 root directory 39
 route 814
 router 797
 routing data 111

routing entries 111
 row 88
 Row Security 429, 431
 rows 84
 RST 460, 469
 RSTAUT 460, 469
 RSTCFG 460, 469
 RSTDLO 460, 469
 RSTLIB 460, 469
 RSTOBJ 460
 RSTUSRPRF 460, 469
 run priority 112
 runtime object 63
 RUNUBE 42

S

SAR 139, 858
 SAV 459, 460, 468
 SAVCFG 459, 460
 SAVCHGOBJ 459, 460, 467
 SAVDLO 459, 460
 Save command 459
 Save file 84
 save-while-active 460
 SAVLIB 459, 460
 SAVOBJ 459, 460
 SAVSECDTA 459, 460
 SAVSTG 459, 460
 SAVSYS 459, 460
 SAW 655
 SBMJOB 111, 114
 scalability 8
 SCOREx 4
 SCPF 115
 secondary thread 116
 Secure Sockets Layer 283
 security 16
 security levels 415
 Security Server 424, 426
 security system values 415
 Security Workbench 429, 430, 431
 self study lab 627
 semaphore 821
 semaphore set 821
 Send TCP/IP Spool File (SNDTCPSPLF) command 386
 Sender Copy 740
 SENTINEL 307
 SENTINEL job 756
 Server Administration Workbench 655
 server AS/400 tuning
 assigning a storage pool 576
 workload/memory, SETOBJACC 575
 server AS/400 tuning, parallel pre-fetch 590
 Server consolidation 19
 server consolidation 20
 server daemon 122
 Server Data Source Master table 557
 server package 65
 Server packages 65
 server storage space 159

Service Director 24, 300
 service pack 244
 service port 556
 service table 123
 service time 561
 SETOBJACC 116, 590
 SETOBJACC command 591
 SETOWAUT 437
 severity 740
 shared memory 821
 shared storage pool 110
 shutdown 308
 sign-on security 421
 Silicon on Insulator (SOI) 14
 single level storage (SLS) 11
 sizing 823
 SLS (single level storage) 11
 SMAPP (System Managed Access Path Protection) 469
 SMB 515
 SMP 28, 87, 589
 SMTARCE 591, 653, 857
 SMTP 115, 787
 SNA Character String 383
 SNADS 771
 SNDTCPSPLF 386
 SNDTCPSPLF (Send TCP/IP Spool File) command 386
 sniffer 660
 soft link 39
 Software Action Request 139, 858
 software requirements 142
 SOI (Silicon on Insulator) 14
 Solution Assurance 839
 source objects 63
 special authority 416
 Specification Table Merge Workbench 488
 spooled file 382
 spooled files 311
 SQL 86, 642, 739
 call level interface 87
 CREATE INDEX 88
 CREATE PROCEDURE 88
 CREATE TABLE 88
 CREATE VIEW 88
 interactive interface 86
 naming convention 88
 package 88
 precompilers 86
 REXX Interface 86
 run SQL statements 86
 run-time 86
 types of SQL statements 89
 SQL (Structured Query Language) 84
 SQL Call Level Interface 87
 SQL commands 313
 SQL index 84
 SQL naming convention - operational difference 635
 SQL package 88, 621
 information 621
 SQL software requirements 630
 SQL view 84

 SSL 283
 SST 747
 Start Host Server command 123
 Start Mail Server Framework Job 115
 Start Printer Writer (STRPRTWTR) command 392
 Start Remote Writer (STRRMTWTR) command 386, 392
 starting OneWorld 307
 starting the SQL Performance Monitor 638
 start-up program 113
 STMT 741
 stopping OneWorld 307, 308
 storage management 11, 38
 Store-and-forward processing 49
 STRCMNTRC 747
 STRCMNTRC command 750
 STRDBG 625, 752
 STRDBMON 607, 616, 625, 766
 stream files 17
 STRHOSTSVR 114, 123
 STRMSF 115
 STRNET 43, 463
 STRPEX command 652
 STRPFRMON command 606
 STRPJ 726
 STRPRTWTR (Start Printer Writer) command 392
 STRRMTSPT 772
 STRRMTWTR 395
 STRRMTWTR (Start Remote Writer) command 386, 392
 STRSRVJOB 751
 STRSST command 747
 STRTCP 114, 115, 787
 Structured Query Language (SQL) 84
 information printing 621
 subnet mask 782, 816
 SUBSTR function 91
 subsystem 110, 113
 subsystem starting 115
 subsystems 110
 super user 771
 support 773, 823, 836
 Support Line 832
 Support Line for Business Partners 840
 swap file 520, 522
 symbolic link 38
 Symmetric Multiprocessing 28, 87
 synchronous I/O 612
 system 60, 462
 System Administration Tools 421
 system ASP 767
 system distribution directory 771
 system information binder 301
 system job 114
 System Licensed Internal Code 468
 System Managed Access Path Protection (SMAPP) 24, 469
 System Message Block 515
 system monitoring 314
 System naming 88
 System naming convention, operational difference 635
 system operator 738, 766

system performance 779
System Service Tools 747
system values 109, 113, 415

T

Tab Security 429
table 84, 88, 629
Table Access Management (TAM) 305, 328
table conversions 537
table index analysis 668
TAM (Table Access Management) 328
Task Dispatch Entries 118
task dispatch queue 118, 563
tax management 846, 847
TCP/IP 7, 115, 291, 747, 773, 781
 CFGTCP 784
 domain name 787
 Host Table Entry 785
 Interface 784
 PING 790
 TCP/IP route 815
TCP/IP configuration 147
TCP/IP Interface 815
TCP/IP performance tips 566
TDE 118
TDQ 118
TechLine 823
Technical Sales Support 839
Technology Independent Machine Interface 7, 9
Technology Solutions Center 824
Telnet 382
Terminal Server 521
Terminal Server Edition 239, 521
test environment 18
thin client 80
thread 112, 116
three-tier design 75
three-tiered application architecture 21
time slice 112, 118
TIMI 7, 9
Tivoli 838
To Program 741
traces 746
transaction 104
transaction boundary 104
transaction processing 103
transaction report, printing 613
transactional data 72
Transmission Control Protocol/Internet Protocol 773, 781
transportation management 847
TRCJOB 752, 754
troubleshooting 760
TSC 824
TSE 239, 521
two-phase commit 105
two-tier design 73, 74
types of SQL statements 89
Typical Plan 189

U

UBE 411
UBE server 71
UDC 433
UDFS 465
Universal Batch Engine Server 71
UNIX 7, 17
update package 64, 328
update process 245
UPDHST 468
upgrade process 246
user ASP 767, 773, 775
User classes 416
User Defined File System 465
user ID 23
user management 23
user name 117
user profile 17, 416, 421, 771
User Profile Revisions 432
user-defined data functions 86
user-defined data types 86

V

V4R4 enhancement 85
verifying host installation 232
Versions 62
view 88
View Results 645
virtual address 12
Virtual Printer Device 391
virtual three-tier design 76
Vision Solutions 532
volume set 520, 521

W

wait 119
Wait-ineligible status 119
waiting 721, 722
Waiting status 118
WAN (Wide Area Network) 73
WCBT 119
WCBTE (Work Control Block Table Entry) 119
Web reporting 847
Web serving 26
WebSphere 26
WebSphere Commerce Suite 27
Wide Area Network (WAN) 73
Windows 2000 290, 519
Windows Terminal Server 142, 285
Work Control Block Table 119
Work Control Block Table Entry (WCBTE) 119
work entries 111
work management 109
 concept 109
 response time curve 563
Work with Active Job 573
Work with Output Queues (WRKOUTQ) command 395
Work with Submitted Job 721
Work with Subsystem Job 721

Workflow Security 433
Workgroup Server 71
workload management 575
worksheet 145
workstation 54
workstation entries 111
Workstation Installation 64, 333
WorldSoftware 4
WorldSoftware Data Dictionary 254
WorldVision 4
worldwide operations, consolidating 21
WRKACTJOB 573, 722, 731, 766
WRKCFGSTS 324
WRKDIRE 771
WRKDSKSTS 572, 766, 775
WRKHDWRSC 771, 782
WRKJOB 719, 731
WRKLNK 736
WRKOBJLCK 720, 728, 766
WRKOUTQ 733
WRKOUTQ (Work with Output Queues) command 395
WRKSBSJOB 721
WRKSMBJOB 721
WRKSPLF 733, 788
WRKSYSSTS 119, 570, 766
WRKSYSVAL 569
WRKUSRJOB 720, 731
WTS 142

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at ibm.com/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number	SG24-5195-01
Redbook Title	J.D. Edwards OneWorld Implementation for AS/400
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	<input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor
Please identify yourself as belonging to one of the following groups:	<input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	<input type="checkbox"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. ibm.com/privacy/yourprivacy/



J.D. Edwards OneWorld Implementation for AS/400

(1.5" spine)
1.5" x 1.998"
789 <-> 1051 pages



J.D. Edwards OneWorld

Implementation for AS/400



**Learn the concepts
and tasks associated
with implementing
J.D. Edwards
OneWorld on AS/400**

**Employ the best
practices for
OneWorld
implementation**

**Make your system run
faster, smoother with
valuable tips and
techniques**

The *J.D. Edwards OneWorld Implementation for AS/400* redbook offers a collection of knowledge gained from J.D. Edwards consultants and AS/400 professionals who have experience with WorldSoftware, OneWorld, and the AS/400 system. This is a detailed guide that explains the concepts, as well as specific tasks associated with implementation of OneWorld on the AS/400 system with *OS/400 V4R4*. This redbook is designed to assist technical people among J.D. Edwards OneWorld customers, OneWorld consultants, business partners, and IBM service representatives. It targets these professionals who are directly involved with the implementation of a total business solution consisting of AS/400 hardware, OS/400, DB2 UDB for AS/400 database, OneWorld solutions, and supplemental solution products.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**

SG24-5195-01

ISBN 0738417300