An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, Communications, and PC Connectivity

Document Number GG24-4200-00

June 1994

International Technical Support Organization
Rochester Center

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xxiii.

First Edition (June 1994)

This edition applies to Version 2, Release 3 and Version 3, Release 1 of Operating System/400, Program Number 5738-SS1 and 5763-SS1 for use with the Application System/400.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Center
Dept. 977, Building 663-3
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.

Abstract

This document describes implementing AS/400 security and how the different security elements fit together. This will help you understand the comprehensive AS/400 security options available to secure your system and data.

It includes information about Auditing, C2 Security, Cryptography, Communications, PC Support, and OfficeVision/400. This document can be used by the new and the experienced AS/400 administrator or auditor. It contains information about security facilities and structures as well as implementation tips and recommendations.

If it is your responsibility to advise AS/400 customers in security issues and provide them with the latest information about Cryptography, C2, Level 50, and the new auditing product, IBM Security/400 (5764-006), then this document is for you.

AB (426 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help auditors and system administrators to develop, implement and audit a solid security system for the AS/400. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/400. See the PUBLICATIONS section of the IBM Programming Announcement for OS/400 and IBM Security/400 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, 208 Harbor Drive, Stamford CT USA 06904.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. 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 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.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
Application System/400
AS/400
CICS/400
ES/9000
IBM
Micro Channel
MVS/ESA
NetView
OfficeVision/400
Operating System/400
OS/2
OS/400
PS/2
SQL/400

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

DEC VT220	Digital Equipment Corporation
UNIX	UNIX System Laboratories, Inc.
SUN, Network File System, NFS	Sun Microsystems, Inc.

Preface

This document is intended to provide a guide to implementing an effective security environment for AS/400 systems. It contains information, tips, techniques, and hints on designing a complete security scheme. Items addressed include the following: physical security, resource security, system level security, performance considerations, and auditing considerations. Where possible, specific implementation examples and coding examples are provided.

The recommendations, tips, techniques, and hints provide system security administrators and security auditors information and tools to assist them in implementing effective security procedures and develop their auditing techniques.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "Overview Of OS/400 and Security Facilities"

  Provides an overview of the AS/400 architecture and security facilities, including terminology, system level protection, object level protection, and physical security.

- Chapter 2, "System Values and Network Attributes"

  This provides an overview of, and recommendations for security related system values and network attributes.

- Chapter 3, "User Profiles and Group Profiles"

  This chapter provides a detailed description of user profiles and group profiles and their parameters. It also discusses auditing at a user level (both user and group profiles), IBM supplied user profiles, and dynamic change of group profiles.

- Chapter 4, "Resource Protection"

  This chapter provides a detailed description of how to protect your resources. Details and recommendations are made on the usage of group profiles, authorization lists, object ownership, spool output, and *SAVSYS special authority.

- Chapter 5, "Authorization Lists"

  This chapter details authorization lists and their usage in a secure environment. It also addresses save and restore considerations with authorization lists.

- Chapter 6, "Audit Journal"

  This chapter provides a detailed description of the security audit journal, QAUDJRN, which is designed to record security related activity.

- Chapter 7, "Security Design For Performance"

  This chapter provides information for designing an AS/400 security environment with regard to performance. It contains recommendations to assist a system administrator in the design of security.

- Chapter 8, "C2 Overview and Considerations"

This chapter provides an overview of the C2 security standard as it relates to the AS/400. It also provides an overview of what steps are required to implement a C2 security scheme on an AS/400 system.

- Chapter 9, “AS/400 Cryptography”

  This chapter provides an overview of cryptography features of the AS/400, including the new hardware cryptographic facility introduced in V2R3.

- Chapter 10, “Communications Security in SNA”

  This chapter provides a description of SNA security related issues with AS/400 in an APPN communications environment.

- Chapter 11, “Security in IBM Communications Applications”

  This chapter provides an analysis of security related issues in several of the common AS/400 communications application environments. Applications addressed include the following: DHCF, NRF, SPLS, DDM, DRDA, DSPT, DSNX, SNADS, TCP/IP, FFS/400, 3270EML, RJE, ECS, OSI and user-written applications.

- Chapter 12, “PC Support”

  This chapter provides details of security related issues unique to PC Support/400. It offers recommendations on how to secure your PC Support environment across various networking environments.

- Chapter 13, “OfficeVision/400”

  This chapter provides a general discussion of security related issues unique to OfficeVision/400.

- Chapter 14, “Auditing the AS/400”

  This chapter provides details on how to use the native AS/400 auditing tools and system commands to assist an auditor or system security administrator to determine the security posture of an individual system or network of systems.

- Chapter 15, “IBM Security/400”

  This chapter provides an overview of the product IBM Security/400 and its function to assist and auditor or system security administrator to determine the security posture of an individual system.

Related Publications

The publications listed in this section are considered suitable for a more detailed discussion of the topics covered in this document. They are divided into V2R3 (and earlier), V3R1, and others.

V2R3 and earlier:

- *Application System/400: Security Reference*
- *Application System/400: Basic Security Guide*, SC41-0047
- *Application System/400 Guide to Enabling C2*, SC41-0103
- *Application System/400 Programming: Basic Backup and Recovery Guide*, SC41-0036
- *Application System/400 Programming: Advanced Backup and Recovery Guide*, SC41-8079

- *Application System/400: System Operator's Guide*, SC41-8082
- *Application System/400 Programming: Control Language Reference*, SC41-0030
- *PC Support/400: User's Guide for DOS*, SC41-8199
- *PC Support/400: DOS Installation and Administration Guide*, SC41-0006
- *PC Support/400: User's Guide for OS/2*, SC41-8300
- *PC Support/400: OS/2 Installation and Administration Guide*, SC41-0007
- *PC Support/400: Technical Reference for DOS and OS/2*, SC41-8091
- *PC Support/400: Application Program Interface Reference*, SC41-8254
- *Application System/400 Communications Management Guide*, SC41-0024 ment
- *Application System/400 Distributed Database Guide*, SC41-0025
- *Application System/400: DDM User's Guide*, SC41-9600
- *Application System/400 Communictions: Distribution Services Network Guide*, SC41-9588
- *Application System/400 Communictions: Remote Work Station Guide*, SC41-0002
- *Application System/400 Communications: 3270 Device Emulation Guide*, SC41-9602
- *Remote Job Entry (RJE) Guide*, SC09-1373
- *Application System/400 Transmission Control Protocol/Internet Guide*, SC41-9875
- *Application System/400 Programming: Work Management Guide*, SC41-8078
- *Application System/400 Programming: System Programmer's Interface Reference*, SC41-8223
- *Application System/400 Programming: Performance Tools/400 Guide*, SC41-8084-01
- *Application System/400 Cryptographic Support/400 User's Guide*, SC41-8080
- *Application System/400 Planning For and Setting Up OfficeVision/400*, SC41-9626
- *Application System/400 Office Services Concepts and Programmer's Guide*, SC41-9758

V3R1 (available during third quarter, 1994):

- *Security - Basic*, SC41-3301
- *Security - Reference*, SC41-3302
- *Security - Enabling for C2*, SC41-3303
- *Backup and Recovery - Basic*, SC41-3304
- *Backup and Recovery - Advanced*, SC41-3305
- *System Operation*, SC41-3203
- *CL Reference*, SC41-3722
- *Client Access/400 DOS Ext Memory Setup*, SC41-3500

- *Client Access/400 DOS Ext Memory User Guide*, SC41-3501
- *Client Access/400 OS/2 Setup*, SC41-3520
- *Client Access/400 OS/2 User Guide*, SC41-3521
- *Client Access/400 DOS, OS/2 Technical Reference*, SC41-3563
- *Client Access/400 DOS, OS/2 API Reference*, SC41-3562
- *Communications Management*, SC41-3406
- *Distributed Data Management*, SC41-3307
- *Distributed Database Programming*, SC41-3702
- *SNA Distribution Services*, SC41-3410
- *Remote Work Station Support*, SC41-3402
- *3270 Device Emulation Support*, SC41-3408
- *Remote Job Entry (RJE) Guide*, SC09-1903
- *TCP/IP Configuration and Reference*, SC41-3420
- *Work Management*, SC41-3306
- *System API Reference*, SC41-3801
- *Performance Tools/400*, SC41-3340
- *Cryptographic Support/400*, SC41-3342
- *Planning for and Setting Up OfficeVision/400*, SH21-0695
- *Office Services Concepts and Programmer′s Guide*, SH21-0703

Others:

- *Centralized Security Administration In An AS/400 Peer Network*, GG24-3719
- *Application System/400 TCP/IP Configuration and Operation Redbook*, GG24-3442
- *The IBM AS/400 as a TCP/IP Network File Server*, GG24-4092
- *VM-AS/400 Connectivity and Functional Use*, GG24-3430
- *Trusted Network Interpretation of The Trusted Computer System Evaluation Criteria, Version 1*, July 1987, NCSC-TG-005, published by the National Computer Security Center.

Additional publications for cryptography:

The following publications must be ordered if cryptography will be used:

- *Application System/400 Common Cryptographic Architecture Services/400 Installation and Operator′s Guide*, SC41-0102
- *IBM TSS Concepts and Programming Guide: Volume I, Access Controls and DES Cryptography*, GC31-3937
- *IBM TSS Programming Reference: Volume I, Access Controls and DES Cryptography*, SC31-2934
- *IBM TSS Concepts and Programming Guide: Volume II, Public-key Cryptography*, GC31-2889
- *IBM TSS Programming Reference: Volume II, Public-key Cryptography*, SC31-2888

- *IBM TSS General Information and Planning Guide*, GA34-2137
- *IBM Workstation Security Services Installation and Operating Guide*, SA34-2141
- *IBM Workstation Security Services Licensed Program Specifications*, GC31-2720

The following manual is considered suitable for a general introduction to AS/400 terminology and concepts:

AS/400 Concepts And Facilities, SX09-1259

International Technical Support Organization Publications

- *Backup Recovery and Media Services/400 Implementation Tips and Techniques*, GG24-4300
- Add others at publication time

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

> *Bibliography of International Technical Support Organization Technical Bulletins,* GG24-3070.

To get listings of redbooks online, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

How to Order Redbooks

IBM employees may order redbooks and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their IBM branch office.

You may order individual books, CD-ROM collections, or customized sets, called GBOFs, which relate to specific functions of interest to you.

Below is a list of ITSO publications (known as redbooks) that are currently available which relate to the AS/400.

AS/400 redbooks are also available on CD-ROM, by adding feature code #8053 to your OS/400 software profile.

- *System/36 to AS/400 System Migration*, GG24-3249-01
- *System/36 to AS/400 Application Migration*, GG24-3250-01
- *AS/400: System/38 Application Migration to AS/400*, GG24-3251-00
- *AS/400 Communication Migration*, GG24-3253--00
- *AS/400 Office in a DIA/SNADS Network*, GG24-3268-00
- *Converting S/36 Environment Application to Native*, GG24-3304-01
- *AS/400 Communications Problem Determination*, GG24-3305-00
- *SQL/400: A Guide for Implementation OS/400 V2R2*, GG24-3321-03

- *AS/400 - S/370 Connectivity*, GG24-3336-00
- *AS/400, S/38 and PS/2 as T2.1 Nodes in a Subarea Network*, GG24-3420-00
- *Writing SAA Applications for AS/400*, GG24-3438-00
- *IBM AS/400 TCP/IP Operation and Configuration*, GG24-3442-02
- *IBM AS/400 in Large Networks: A Case Study*, GG24-3447-00
- *AS/400 Communications Definitions Examples*, GG24-3449-00
- *AS/400 Object Distribution Facility and SNA RSCS PROFS*, GG24-3479-00
- *IBM AS/400 ISDN Connectivity*, GG24-3517-00
- *OfficeVision/400 and AS/400 Query Applications in a Multilingual Environment*, GG24-3579--00
- *AS/400 V2R3 Systems Management*, GG24-3614-02
- *OfficeVision/400 in a DIA/SNADS Network*, GG24-3625-00
- *PC Support/400 Implementation and Performance*, GG24-3636-00
- *AS/400 Audit and Security Enhancements in OS/400*, GG24-3639-00
- *WAF/400 5363 Optical Subsystem Configuration and Installation*, GG24-3680-00
- *OfficeVision/400 Printing*, GG24-3697-00
- *AS/400 Printing II*, GG24-3704-00
- *AS/400 APPN with PS/2 APPN, 3174 APPN, 5394 and Subareas*, GG24-3717-00
- *AS/400 CPI Communications Selected Topics*, GG24-3722-00
- *AS/400 Performance Management V2R2*, GG24-3723-01
- *Multimedia Examples with the AS/400 Using AVC*, GG24-3743-00
- *Getting Started with AS/400 OSI*, GG24-3758-00
- *AS/400 Communication Definition Examples Volume 2*, GG24-3763-00
- *Installation Considerations for National Language*, GG24-3790-00
- *Artificial Intelligence and AS/400: Neural Networks and Knowledge Based Systems*, GG24-3793-00
- *Facsimile Support/400 Implementation*, GG24-3797-00
- *Application Development on the AS/400*, GG24-3806-00
- *PC Support/400 Asynchronous and SDLC Configuration Examples*, GG24-3808-00
- *5494 & OS/2 ES: Connecting Remote User Groups*, GG24-3828-00
- *AS/400 Automation Using NetView and SNA MS Transport*, GG24-3841-00
- *DOS PCS/400 in OS/2 V2 Virtual DOS Machine*, GG24-3856-00
- *WAF/400 Administration and User Examples*, GG24-3866-00
- *OfficeVision/400 Application Enabler*, GG24-3868-00
- *Cooperative Processing and GUI in an AS/400 Environment*, GG24-3877-00
- *OfficeVision/400 Application Programming Interfaces V2R2*, GG24-3885-00
- *OfficeVision/400 Integration with CallPath/400 and Fax Support*, GG24-3896-00

- *AS/400 Performance Capacity Planning V2R2*, GG24-3908-00
- *AS/400 System Availability and Recovery for V2R2*, GG24-3912-00
- *AS/400 Network Routing Facility*, GG24-3918-00
- *AD/CYCLE Code/400, ADM/400 and ADS/400*, GG24-3928-00
- *OfficeVision/400 V2 Technical Tips and Techniques*, GG24-3937-00
- *CICS/400 Migration from Mainframe CICS*, GG24-4006-00
- *Using DOS PC Support/400 with Novell NetWare 3.11 and NetWare for SAA 1.3*, GG24-4013-00
- *Ultimedia Video Delivery System/400*, GG24-4020-00
- *AS/400 Client Series - Products and Positioning*, GG24-4027-01
- *IBM AS/400 Printing III*, GG24-4028-00
- *Performance Benchmarking for the AS/400*, GG24-4030-00
- *AS/400 and RISC System/6000 Connectivity*, GG24-4039-00
- *Current-OV/400 Workgroup Program*, GG24-4050-00
- *Using V2R3 DOS and OS/2 PC Support under OS/2 2.1*, GG24-4070-00
- *Apple Macintosh and the AS/400*, GG24-4071-00
- *OfficeVision/400 Application Enabler Version 2 Release 3*, GG24-4072-00
- *The IBM AS/400 as a TCP/IP Network File Server*, GG24-4092-00
- *ENVY/400 Hints and Tips*, GG24-4094-00
- *Introduction to ENVY/400*, GG24-4126-00
- *AS/400 Integrated Language Environment*, GG24-4148-00
- *CICS/400 V2R3 Task Book*, GG24-4182-00
- *AS/400 V2R3 Software Life Cycle Mgmt with ADM/400*, GG24-4187-00
- *OfficeVision/400: Printer Setup in an OfficeVision Environment*, GG24-4283-00
- *AS/400 Client Series Handbook*, GG24-4285-00
- *BRM Services/400 Implementation Tips and Techniques*, GG24-4300-00

Acknowledgments

The advisor for this project was:

Craig Tamlin
International Technical Support Center, Rochester Center

The authors of this document are:

Wayne O. Evans, AS/400 Security Consulting & Education
Harri Levänen, IBM Finland
Hiro Nishihara, International Technical Support Center, Rochester
Bent Philipson, IBM Denmark
Jorge Salvador B., IBM Ecuador
Sverre Selbach, IBM Norway
Sue Van Riper, IBM Rochester I/S
Jeff Wang (Wang Jiunn Jye), IBM Taiwan

The authors of the previous version documents are:

Örjan Jönsson, IBM Sweden
Martin Kotlenga, Coopers and Lybrand, USA
Fred Shuback, Coopers and Lybrand, USA
Pia Tøt, IBM Denmark
Jay Ehrenreich, Coopers and Lybrand, USA
J. R. Mattiucci, Coopers and Lybrand, USA
Dan Naidoo, Information Services Management, South Africa
Santiago Moya, IBM Argentina
Nik Ariff N Omar, IBM Malaysia
Bernd Wagner, IBM Germany

This publication is the result of a residency conducted at the International Technical Support Center, Rochester Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Carol Woodbury, IBM Application Systems, Rochester MN.
Paul Brown, IBM Application Systems, Rochester MN.
Janet Krueger, IBM Application Systems, Rochester MN.
Dick Locker, IBM Application Systems, Rochester MN.
Jelan Heidelberg, IBM Application Systems, Rochester MN.
Terry Hennessy, IBM Application Systems, Rochester MN.
John Griffith, IBM Application Systems, Rochester MN.
Jeff Uehling, IBM Application Systems, Rochester MN.
Aleesa Kobi, IBM Application Systems, Rochester MN.
Kathy Konkel, IBM Application Systems, Rochester MN.
Jim Powell, Keane Inc., Rochester MN.

Chapter 1. Overview Of OS/400 and Security Facilities

The Application System/400* (AS/400*) has a very sophisticated and modern architectural base. While it has evolved from the System/36 and the System/38, it contains many facilities that AS/400 customers are only now utilizing to the maximum capability.

Like the AS/400, its operating system, Operating System/400* (OS/400*), has evolved from the S/36 and S/38, and offers a very sophisticated and powerful computing base.

This chapter is an overview for readers who are not regular AS/400 users. It offers an overview of OS/400 architectures and relevant object structures, and then highlights OS/400 security features.

1.1 Basic AS/400 Terminology

This section introduces basic AS/400 terminology. Understanding these terms and concepts are essential to understanding AS/400 security. For a more information on these, refer to the *AS/400 Concepts And Facilities* manual.

1.1.1 System Object Structure

There are a few basic concepts that will enable you to quickly start work with the AS/400 system. These are described below.

Objects

Everything on the system that contains some form of information and can be accessed via the standard OS/400 interface is represented as an *object*. An object is made up of a set of attributes that describe the object and some form of data. The attributes of an object include its name, type, size, the date it was created, a short description, and the name of the library in which it is stored. Every object on the system has an *owner*, who plays an important part in the security functions. The owner of an object can grant or revoke access of other users. A user cannot be deleted before all objects that are owned by the user are deleted or assigned to another user.

The data component of an object is the collection of information that is stored in the object. The data part of a program is the instructions that make up the program. The data portion of a file is the collection of records that make up the file. The term ″object″ is used to refer to a number of different items that can be stored in the system, regardless of what the items are.

Object Types

Different object types have different operational characteristics. These differences make each object type unique. For example, because a file is an object that contains data, its operational characteristics differ from those of a program, which contains instructions.

Objects are arranged with a common object header and a type-dependant functional portion. For example, a program object header would contain a description of the object, including the type (program), and the owner (the user who created the program). The functional part of the program object contains information that governs the way the object can be used. This allows the system to perform operations collectively on all objects, as well as permitting each object to be tailored for its own purposes. Figure 1-1 on page 1-2 shows the common object structure.

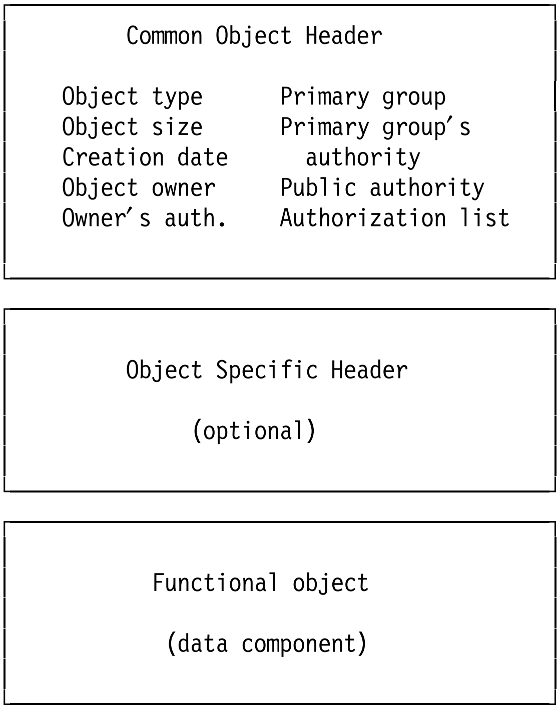


Figure 1-1. OS/400 Object Structure. The structure of an object is made up of three primary parts: the common object header, the object specific header, which is optional, and the functional object (data component).

There are over 60 different object types. Some examples are: files, commands, documents, programs, libraries, controller descriptions, device descriptions, user profiles, and job queues. Each object type has an abbreviated form preceded by an asterisk, to signify this as a reserved word. For example a file is referred to as an object of type *FILE. For the purposes of security, each individual object may be secured, tailoring user access to the function of the system.

In addition, certain object types may have many attributes which further differentiate the types. For example, an object type of program (*PGM) could have an attribute of RPG, CBL, C, or CLP, for instance. An object type of file (*FILE) could have an attribute of PF, LF, DSPF, DDMF, or SAVF, and so on. Each of these attributes allow deeper levels of sub-typing for particular specialized function.

Libraries

A library is an object that is used to group related objects and to locate objects by name. Thus, a library is like a directory to a group of objects. Libraries can be used to group the objects into any meaningful collection. Objects can be grouped according to security, backup, processing, or any other user requirements.

Some libraries come with the AS/400 system and are standard parts of the system. The QSYS library contains the base operating system components, as well as other libraries. It is special in that it is the only library that can contain other libraries. Users can create other libraries. IBM Program Products usually have their own libraries. Users should avoid adding objects to IBM libraries.

Library access rules govern the use of commands, programs, files or any object you may want to use. You must have access to the library as well as the object you want to use. The system will look for the object in the libraries specified in the library search path (called a library list, abbreviated to *LIBL), or you can refer to an object by specifically stating a library and the object name.

Object Names

Each object has a name. The object name and the object type are used to identify an object. The object name is explicitly assigned by the system for system-supplied objects, or by the user when creating an object. The object type is determined by the command used to create the object. Several objects can have the same name so long as their object types differ, or as long as they exist in different libraries.

Object names can be qualified using the name of the library where that object exists. The combination of the object name and the library name is called the *qualified name* of the object. An *unqualified name* contains only the name of the object, and is useful only when the object and type are unique on the system or within the object′s library list.

The object name in the following examples is always MASTFILE; the different library names and object types identify the following objects uniquely:

```
ACCTPAY/MASTFILE *FILE - file MASTFILE in library ACCTPAY
ACCTPAY/MASTFILE *PGM  - program MASTFILE in library ACCTPAY
ACCTPAY/MASTFILE *MENU - menu MASTFILE in library ACCTPAY
ACCTREC/MASTFILE *FILE - file MASTFILE in library ACCTREC
```

These four objects may all exist at once without naming or type conflicts.

Document Library Objects

Within the OfficeVision/400 and PC Support/400 products, two main object types are used: documents (*DOC) and folders (*FLR). Collectively these are termed *Document Library Objects* (DLOs). These are most commonly created using OfficeVision/400 or PC programs accessing the AS/400 with the PC Support/400 product, although they can be created and used outside of these products.

A folder is an object that is used as a directory for documents or other folders. For example, a user of a personal computer could store personal computer programs, files and documents created with his PC programs in folders on an AS/400 system.

Folders can be filed within another folder. Folders within folders can be considered to be like drawers in a filing cabinet. The filing cabinet itself is the root directory, (as it is termed on a PC), and the drawers are considered as sub-directories. The AS/400 folder structure was designed like the directory structure on a PC, so that PC users could easily use AS/400 folders for storing their objects. These objects can be shared with OfficeVision/400 users, as this same folder structure is used to store OfficeVision/400 documents.

Users can access the documents and folders through the PC Support and OfficeVision/400 interface. However, all documents and folders are actually located in the IBM supplied library QDOC. The library name QDOC is never specified by the user when accessing documents and folders; this is simply a storage mechanism.

Document Library Object Names

A document or folder name can be 1 to 12 characters long, including an optional extension. If no extension is included, a document or folder name can have a maximum of 8 characters. If an extension is included, the extension must start with a period and can have up to 3 additional characters.

Folder names should not begin with a Q because the IBM-supplied folder names begin with Q. The following are examples of permitted names for Document Library Objects:

```
LTR001.DOC
LETTERS/LTR001.DOC
PAYROLL/LETTERS/LTR001.DOC
APPMAIL/PAYROLL/LETTERS/LTR001.DOC
```

The “/” is used to separate folder names in the path and the document name.

Files

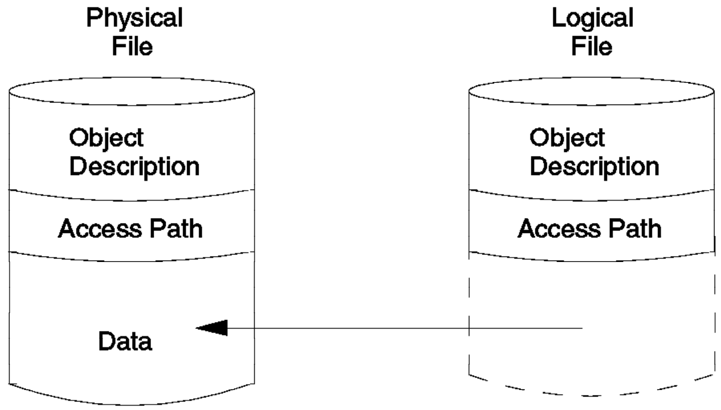
Files are commonly secured objects on any computer system. On the AS/400, there are twelve categories of files:

PF	Physical files (for data records and program source)
LF	Logical files (for relational database functions)
SAVF	Save files (for online save/restore)
DDMF	Distributed Data Management (DDM) files
DSPF	Display file (for I/O to the screen)
PRTF	Printer file (spool file definition)
DKTF	Diskette (for I/O to diskette)
TAPF	Tape files (for I/O to tape)
CMNF	Communications files
BSCF	BSC communications files
MXDF	Mixed files
ICFF	ICF communications files

Typically, discussions of security are related to physical and logical files, as these are used by application programs and contain application data. Most often, the other file types do not present the same security exposure.

Physical Files: A *physical file* contains both the description of the data fields of the file, and the data itself. The record description is maintained in a separate portion of the object from the data. Before any records can be written in a file, the file must be created with the record description.

Logical files: *Logical Files*, or views, as they are called in SQL/400*, do not actually contain data, but describe how records contained in one or more physical files are to be presented. They define an alternate record layout or access path for the file.



Description contains:
- Common Object Information (including security attributes)
- Field Names
- Field Headings
- Other Field Attributes

Access Path contains:
- Key Sequence Details

Data:
- The Records

Description contains:
- Different Object Information and security attributes
- Different Field Names
- Different Field Headings
- Different Field Attributes

Access Path contains:
- Structures to support the five relational operators:
  - Sequence
  - Project
  - Select
  - Union
  - Join

Data:
- Accessed from underlying Physical Files

Figure 1-2. Physical and Logical Files. Physical files have three main parts: a description, an access path and data. Logical files have two parts: a description, an access path, and it also contains pointers to data fields that are in physical files. Programs using logical files treat them exactly the same as if they were physical files.

File Members: A physical file may contain *members*, which are groups of records. Records are made up of many fields. Database files - files with transaction and master file data - often consist of only one member. In contrast, a file holding source program statements usually contains several members, one per source program.

Authorities for a file apply to all the members in the file. The members are not seen as having individual authorities. However, individual authorities are actually stored with each member. All these ″extra″ authorities can affect the performance of system save and restore operations, but they have no impact on the normal user. There are no commands to individually manipulate the member authorities.

Object Structure Summary

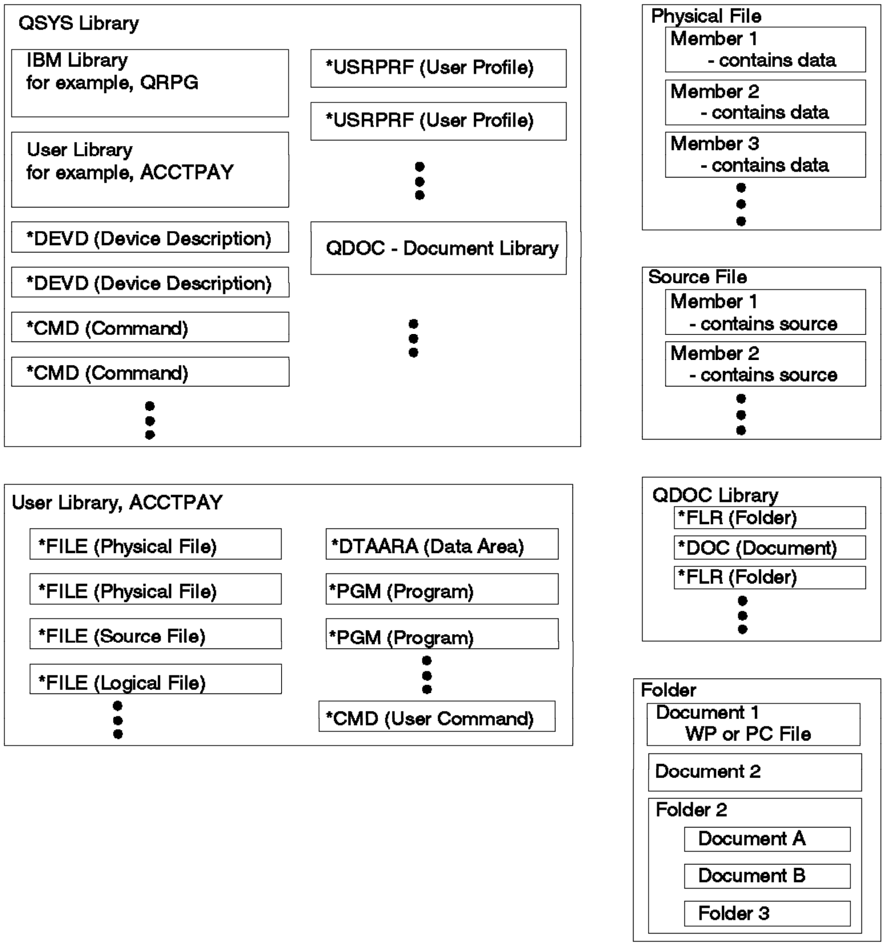


Figure 1-3. Object Structure Summary. This figure shows how the previously mentioned objects relate to each other as they reside on the AS/400 system.

1.1.2 Work Management

Work management supports the commands and operating system functions necessary to control system operations and the daily workload on the system. It controls resources for applications, such as workstations, and storage, so that the system can support multiple applications and system tasks.

All the work done on the system is submitted though the work management functions. When OS/400 is installed, it includes a work management environment that supports interactive, batch, and communications jobs. The operating system can be tailored to create an individual, user-defined work management environment.

Jobs

The AS/400 system uses the term *job* to refer to your workstation session as well as any ″batch″ jobs or ″system″ jobs that may be in the system. There are five types of jobs relevant to security:

- Interactive job
- User or operator submitted batch job
- Communications job
- Autostart job
- Prestart job

The security implications of each of these are discussed below.

Interactive Job: An interactive job is started when a user signs on to a work station. That is, a workstation session is called an interactive job. The user is identified to the system with a *User Profile*, and the authentication is tested through password checking (at security levels 20 and above).

Batch Job: A workstation user or operator can submit a batch job. The user profile under which the batch job executes can be the same as the profile of the submitting user, or it can be a different user profile. In order to use a different user profile, the submitting user must be authorized to use a *job description*, an object containing the job execution attributes, which includes the other user profile name. Alternatively, the user can submit a job under a different user profile by specifying a profile name on the user profile parameter of the Submit Job command. In both cases, the submitter must be authorized to the user profile. These techniques allow good control over job submission on behalf of other users.

Communications Job: A communications job is started when another system issues a request over a communications line. Many techniques are available to control the attachment of a proper user profile to that job. These are discussed in detail in Chapter 10, “Communications Security in SNA” on page 10-1.

Autostart Job: This type of job is started automatically when a subsystem is started. It requires a job description to identify the user profile for the job. An autostart job can be used to perform some operations on a routine basis. For example, the QBASE and QCTL subsystems have autostart jobs that start up printer spooling.

Prestart Job: This type of job is used for communication purposes. When a program A on a remote system wishes to communicate with a program B on the local system, program A must send a request to load program B. This takes time. To speed up the load process program B can be loaded in advance by defining it as a prestart job. A prestart job is started when its subsystem is started.

Job Queues

A *Job Queue* (*JOBQ) is a list of jobs waiting to be run by the system. Each job queue is associated with a *subsystem* (the processing environment). A job is placed on a job queue by the SBMJOB command or by starting a spool reader that reads the job from a diskette or database file. Jobs are selected from the job queue to run based on the job queue scheduling priority. Security information can be included in the job queue description to define who can control the job queue and manage the jobs on the queue.

Subsystems

A *subsystem* (*SBSD) is a single, predefined operating environment through which AS/400 work flow and resource use are coordinated. A subsystem is a means to separate activities on the system - for example, interactive users and batch jobs. Each piece of work running in a subsystem is called a *job*. In a subsystem, *work entries* are defined to identify the sources from which jobs can be started for running in that subsystem. A *Communications Entry* is an example of a work entry. Devices are simply a source of work for a subsystem. Each work entry defines one or more devices or remote locations that are controlled by the subsystem. The devices are allocated by the subsystem for receiving program start requests for the jobs.

Regardless of how a job is started it must use a job description (JOBD). Jobs are processed as one or more consecutive *routing steps*. A routing step is the processing done as a result of a call to a program specified in the subsystem′s *routing entry*. When a job is started, the correct routing entry is selected by means of *routing data*. Routing data is extracted from the job description for the job.

A single AS/400 may have many subsystems defined. There are several subsystem configurations shipped by IBM with OS/400 and with additional licensed products. AS/400 software vendors may supply software packages with subsystem descriptions. Customers may also define their own subsystems.

The QBASE subsystem provides a single combined environment for interactive, batch, and communications jobs and also provides the subsystem control function. It is typically used with less complex environments. Most often, customers with a mixed workload will choose to separate the workload and use other subsystems - either shipped by IBM (such as QCTL) or defined by the user. Subsystems provided by IBM are summarized in Table 1-1. All of these subsystems may not be active at the same time.

Subsystem	*Function*
QBASE	Control function, interactive, batch and communications jobs
QBATCH	Batch subsystem
QCMN	Communications subsystem
QCTL	Controlling subsystem. Supports only the system console
QINTER	Subsystem for interactive work
QPGMR	Programmer subsystem
QSNADS	For SNA Distribution Services
QSPL	Spooling subsystem
QSYSSBSD	Backup subsystem - used if controlling SBSD is damaged
QSYSWRK	Operating system jobs
QDSNX	For Distributed Systems Node Executive
QFNC	For Finance communications
QTCP	For TCP/IP communications

Table 1-1. Subsystems Shipped by IBM. Other subsystems may be created for installation needs or supplied with packaged applications.

Output Queues

An *Output Queue* (*OUTQ) is a list of spooled files waiting to be printed. Printer files are objects used to define the attributes for output from jobs on the system. If the processing of a job results in output, the subsystem running the job creates the output as one or more spooled files in the output queue. Subsystems themselves have output associated with starting and completion status. Security for output queues is considered in 4.9.1, “Security in Output Queues” on page 4-9.

Message Queues

A *message queue* (*MSGQ) is a list in which messages are placed when they are sent to a user or program. A message queue is like a mailbox for messages. Messages sent to an ″address″ in the system (such as a user or a program) are put in the message queue associated with the address.

System operator message queue: The QSYSOPR message queue is a special message queue to which the system sends messages regarding changes in the status of the system, devices, and jobs, and messages indicating a condition that needs operator intervention.

1.1.3 Configuration Descriptions

Configuration Descriptions are objects used to define the characteristics and arrangement of devices and communications links attached to the AS/400. Configuration descriptions are linked together to form a hierarchy:

Networks (***NWID**)

Lines (***LIND**)

Controllers (***CTLD**)

Devices (***DEVD**) printers, displays, tape, diskette, communications

Modes (***MODD**)

Classes of Service (***COSD**)

The AS/400 has the capability to create certain configuration descriptions automatically such as locally attached controllers and devices. This is called *autoconfiguration*. This significantly reduces the initial installation tasks that need to be performed. The default authority given to configuration objects is such that any user can use it, for example, to sign on to a locally attached display station.

1.1.4 Programs and Commands

AS/400 terminology differentiates between programs and commands. A *command* is used to request a function of the system. A command consists of a command name, indicating the type of action to be performed, together with optional parameters, defining more detail about the command. For example, to create a user profile, enter the command CRTUSRPRF and specify the parameters to create the required user characteristics. Users can create their own commands, using the ″create command″ command (CRTCMD). A command invokes program code, called the *command processing program*.

A *program* means a user or vendor program, written in a language such as CL, RPG, COBOL, or C. Program source statements are created as members of a source file. The members are compiled (using one of the create program commands) a process that creates the program as a new object (*PGM). Like commands, a program can also be invoked from a workstation, but requires a CALL command to invoke it.

Since commands and programs are objects (*CMD and *PGM), they are subject to normal AS/400 resource security.

Control Language (CL)

The commands mentioned previously make up the primary user interface on the AS/400, which is called Control Language (CL). This is a large, rich and functional set of commands that are used in every environment including programming, communications, configuration, performance management, messaging, save and restore, printing, problem management, office, online education and security. These commands can be combined to make up a program (called a CL program). Some applications can consist of only CL programs.

Application Program Interface (API)

An API is a functional interface supplied by the operating system or a separately orderable licensed program. It allows an application program in a high-level language to use specific data or functions of the operating system or licensed program.

1.2 AS/400 Security Architecture

System security is an integrated function of the AS/400 system. It is implemented at the instruction level 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, including program development and execution, data base 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.

1.2.1 S/36 and S/38 Compatibility

AS/400 is the follow-on system for the S/36 and the S/38, and has been designed to be compatible with both architectures. Similarly, security features from both environments have been combined in AS/400 system security, with significant enhancements. Combining both original security facilities results in some functional redundancy that is necessary to accomplish easy and secure migrations to AS/400. It is possible to operate an AS/400 in a S/36 or S/38 *environment* in which the user command interface (the screen menus) are similar to the prior system. This does not, however, affect the underlying security functions of the AS/400.

1.2.2 System Integrity

The integrity of the operating system is an important prerequisite for the implementation of security controls. The AS/400 system has good integrity for several reasons:

- Precisely controlled storage addressing limits for a user
- Security implementation at the instruction level
- A physical keylock controlling the operating system security environment
- A precisely defined method for providing limited capabilities for users
- A security system that is an integral part of the total system
- A communications environment with security features built in at the lowest level
- Special hardware to validate software pointers
- Complete auditing capabilities of system and user functions

1.2.3 Single-Level Storage

Storage allocation on the AS/400 is very different than on most other computer systems. The AS/400 uses a shared storage system in which all portions of main and auxiliary storage are addressed as though they are within a single area (or level). The system uses the object name to determine where it exists in storage. This means that the user can find objects by name, rather than by storage locations. Because operations cannot be performed on an object that is not in main storage, the system moves all or part of the object into main storage as it is needed, and moves it back into auxiliary storage when the object is not needed. This transfer is controlled by the system, and does not require control by the user or programmer.

Because objects can be accessed only by name, security cannot be bypassed to access an object directly.

1.3 AS/400 Security System Values

The AS/400 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

System Values are covered in detail in Chapter 2, “System Values and Network Attributes” on page 2-1.

1.4 AS/400 Users and Groups

The following terms and concepts are involved in defining users and their authorities to the AS/400. Users are defined with profiles; the users can:

- Be organized into groups
- Have special capabilities
- Have special limitations

1.4.1 User Profiles

User Profiles contain information describing a system user, that user′s privileges and limitations when using the system, and lists of objects 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. Examples of the security elements of User Profiles are given in Chapter 3, “User Profiles and Group Profiles” on page 3-1.

1.4.2 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. These special authorities are not hierarchical.

***ALLOBJ** All object authority is granted for accessing any system resource

***AUDIT** Allows the user to perform auditing functions

***JOBCTL** Allows manipulation of job and output

***SAVSYS** Used for saving and restoring the system and data without having explicit authority to objects queues and subsystems

***SECADM** Allows administration of User Profiles and Office

***SERVICE** Allows access to special service functions for problem diagnosis

***SPLCTL** Allows control of spool functions

***IOSYSCFG** Allows change of system configuration

1.4.3 User Classes

There are five *user classes* which are hierarchical in authority. The classes represent different roles in the DP environment. These are convenient ways to assign the *special authorities* listed above to different types of users. A higher class can perform all the functions of a lower class; for example, *SECOFR includes the privileges of *SECADM by default. The following are the five user classes:

***SECOFR**	Security Officer
***SECADM**	Security Administrator
***PGMR**	Programmer
***SYSOPR**	System Operator
***USER**	End User

The user class also affects what options are shown on the system menus. A user with higher authorities will see more of the system menu options. A user with less authorities will only see the menu choices allowed by the user class. A user may be given any of the special authorities regardless of his 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.

1.4.4 IBM-Supplied User Profiles

The AS/400 has a number of user profiles provided as part of the operating system. Only QSECOFR is intended for signon. Others are for ownership of various objects and system functions.

```
QDBSHR
QDFTOWN
QDOC
QDSNX
QFNC
QGATE
QLPAUTO
QLPINSTALL
QMSF
QNETSPLF
QPGMR
QRJE
QSNADS
QSPL
QSPLJOB
QSRV
QSRVBAS
QSYS
QSYSOPR
QTCP
QTSTRQS
QUSER
```

1.4.5 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. In V3R1 a user may be a member of up to 16 different groups. In earlier releases the user can only be a member of one group. In addition, only one level of grouping is permissible. For example, if user profile FRED belongs to group profile DEPTA, DEPTA cannot belong to another Group Profile.

Group profiles are used to organize users along job functions and to 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 and that good naming conventions are used.

A group profile is implemented as a user profile; that is, it is created just like a user profile, and when granting authority, the AS/400 does not treat groups any differently than user profiles. The two uses may be intermixed. 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 any sign on to the profile.

1.4.6 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. Some local programming (or the use of 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 will affect initial program, initial menu, current library, the current attention program (associated with the attention key on the terminal), and access to general system commands.

1.5 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.

1.5.1 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 as such. It is possible to have authority for these operations without having access to the data stored in the object. Likewise, one can have full access (read, write, update, delete, execute) to the data in an object without having full authority to manipulate the whole object.

The following diagram shows the relationship between the object rights and the data rights.

```
                       Object Rights
                    ┌─────*OBJOPR    (Object Operational)
                    │     *OBJMGT    (Object Management)
      Object        │     *OBJEXIST  (Object Existence)
                    │     *AUTLMGT   (Authorization list management)
     Authority      │     *OBJALTER  (Object Alter)
                    │     *OBJREF    (Object Reference)
                    │
                    └──►Data Rights
                          *READ      (Read data)
                          *ADD       (Add data)
                          *UPD       (Change data)
                          *DLT       (Delete data)
                          *EXECUTE   (Run a program)
```

*Figure 1-4. Object Authority Elements. All object authorities are made up of some combination of these elements. For example, *USE authority is really *OBJOPR and *READ.*

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 capability 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 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 data base files and change the attributes of SQL packages.

OBJREF: This authority is needed to specify a data base 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. To delete the whole 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 have been given special names as an abbreviated form. For example, *USE is the combination of *OBJOPR, *READ, and *EXECUTE.

***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

1.5.2 *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 group(s) has no specific authority to the object.

Public authority is explained in more detail in Chapter 4, “Resource Protection” on page 4-1.

1.5.3 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.

1.5.4 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.

1.5.5 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 following are some of the events that may be logged:

- Authorization failures
- Object creations
- Object deletions
- Changes to jobs
- Move or rename of objects

- Changes to system distribution directory or office mail actions
- Obtaining authority from programs which 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
- Users′ access to audited objects
- CL command strings

Information from the audit journal can be extracted into a database file, then examined by an auditor using a tool such as Query/400 to locate security violations or exposures.

Security/400 has programs that produce reports from the audit journal receiver. See Chapter 15, “IBM Security/400” on page 15-1.

1.5.6 Authority Holder

An *authority holder* is an object that specifies and reserves an authority to a program-described database file before the file is created. When the file is created, the authority specified in the holder is linked to the file. The authority holder is for use mainly in the System/36 Environment.

1.6 Physical Security

Physical and procedural security controls provide the basis on which other controls such as software security are built. In addition to physical access control and output distribution procedures, which are necessary controls in any computing environment and therefore not mentioned here, the AS/400 has two unique hardware features, which are important for physical security:

- System Keylock - to enable or disable certain system service functions
- Display Station functions - keylock, and play/record keys

These are discussed in more detail in 14.5.1, “Physical Security” on page 14-8.

1.6.1 The History Log (QHST)

The *history log* (QHST) contains a subset of messages that are sent about system operational events to the system operator message queue.

Some messages relating to system security are written in the system history log. However, this function is now superseded by support offered by the security audit journal. QHST should not be used as a source for tracking security-related events as it may have been in the past.

1.6.2 Cryptographic Support

The IBM Common Cryptographic Architecture Services/400 PRPQ is used for hardware cryptography on the AS/400. The hardware encryption feature is available in two versions. The full function version has a restriction on who is allowed to use it. The “commercial” function is without such restrictions. Refer to “Export Restrictions for DES” on page 9-2 for more information. The product currently provides the following:

- Data Encryption Standard (DES)
- Public Key Algorithm (PKA)
- Key Management functions
- Security Application Programming Interfaces (SAPIs) to invoke cryptographic functions from user-written applications
- Security for the IBM 4700 banking terminal

The Security Application Programming Interface (SAPI) verbs must be used to implement file cryptography. A system command or utility program for the encryption or decryption of files is not currently available.

The IBM Common Cryptographic Architecture Services/400 is capable of exchanging encrypted data with any products that implement DES.

Chapter 2. System Values and Network Attributes

This chapter describes security-related system values and network attributes. This includes:

- A description of each system value and network attribute
- A table with recommended values
- A discussion about the security levels 40 and 50 and how to implement them in your system.

Note: The intention of this chapter is to give an overview with tips and recommendations. Refer to the *AS/400 Security Reference* for detailed information.

2.1 System Values

The first topic describes the system values that control security on your system. The security system values are broken into four main groups:

- General system values
- Other system values related to security
- System values that control passwords
- System values that control auditing

2.1.1 General Security System Values

The system values listed below can be changed through the Change System Value (CHGSYSVAL) command or using the Work with System Values (WRKSYSVAL *SEC) command. Changes to the system values become effective immediately, except for the security level (QSECURITY) value, which becomes effective only after the next IPL.

QALWOBJRST Allow objects that are security-sensitive to be restored to the system.

Specifies whether system state objects or objects that adopt their owner′s authority may be restored to the system.

QALWUSRDMN Allow user domain objects in the libraries.

Specifies which libraries are allowed to contain user domain objects of type *USRSPC, *USRIDX, and *USRQ. These objects are a potential security exposure on a system with high security requirements. The system cannot audit the movement of data to and from user domain objects. To learn more about user domain objects, refer to “Special Object Attributes and Types” on page 2-8.

QALWUSRDMN can be left at its default value at security levels below 40. It must be considered when going to level 40 or higher.

QCRTAUT Authority for New Objects.

This value is used to determine the public authority of a newly created object, if the following conditions are met:

- The create authority (CRTAUT) parameter for the library of the new object is set to *SYSVAL.
- The new object is created with public authority (AUT) of *LIBCRTAUT (the default).

The default value is *CHANGE. It is recommended that you do not change this value. It is better to change the CRTAUT value at the library level. It may impact your day-to-day operations.

QDSPSGNINF Display Signon Information.

Specifies that the signon information display is to be shown. This displays information such as the date of last signon, invalid signon attempts, and the number of days until the password expires (if applicable).

This information can alert users that there has been unauthorized attempt to access the system using their user profile.

For users requiring a value different from the system value, the DSPSGNINF keyword for an individual user profile can be set to *YES (to display the information) or *NO (for no information displayed).

QINACTITV Inactive Job Time-Out Interval.

Specifies in minutes how long the system allows a job to be inactive before taking action. A workstation is considered to be inactive if is waiting at a menu or display, or if it is waiting for some message input with no user interaction.

When you specify a time-out interval, if a job reaches that interval the system will take the action specified in the QINACTMSGQ system value. Local jobs that are currently signed on to a remote system are excluded. PC Support/400 jobs are also included; you can find more information about this in Chapter 12, “PC Support” on page 12-1.

An inactive workstation might allow unauthorized persons access to the system. This system value helps you to prevent users from leaving workstations inactive.

Be sure to discuss the impact of a change of QINACTITV with the users on the system and inform them at the time you make the change.

QINACTMSGQ Inactive Job Time-Out Message Queue.

The QINACTMSGQ value specifies either the name of the message queue to which a notification message is sent, or the action the system takes when an interactive job has been inactive for a specified interval of time. The time interval is specified by the system value QINACTITV.

There are considerations for PC Support/400 jobs. They are explained in Chapter 12, “PC Support” on page 12-1.

QLMTDEVSSN Limit Device Sessions.

Specifies whether users are limited to sign on to more than one device at one time.

QLMTSECOFR Limit Security Officer.

Restricts privileged users (with *ALLOBJ or *SERVICE authority) to specified workstations.

A privileged user who leaves the terminal unattended represents a considerable security exposure.

QMAXSIGN Maximum Number of Signon Attempts.

Defines the maximum number of invalid signon attempts by local or remote users. This also works for PC Router signon. Invalid attempts are any combination of invalid password, invalid user profile, or inadequate authority to the display station.

Once a user has reached the maximum attempts value, the system will take the action specified in QMAXSGNACN.

The value should be high enough to allow correction for typing errors but low enough to prevent opportunities to guess a valid user profile and password.

You can use security auditing to log signon violations. See Chapter 6, “Audit Journal” on page 6-1. You must create a query, or you can use Security/400. For more information on Security/400 refer to Chapter 15, “IBM Security/400” on page 15-1.

QMAXSGNACN Action When Signon Attempts Reached.

This system value determines what the system does when the maximum number of signon attempts (specified in QMAXSIGN) is reached.

Possible values for QMAXSGNACN are:

- 3: Disable both the user profile and device.
- 1: Disable the device only.
- 2: Disable the user profile only.

With, PC Support/400, invalid attempts will only disable the user profile, but not the device. Refer to Chapter 12, “PC Support” on page 12-1 for further details.

If you create the message queue QSYSMSG in QSYS, messages about critical system events are sent to that message queue as well to QSYSOPR. You can use the QSYSMSG message queue to monitor any invalid attempt to signon to the system, just by seeing it or controlling it by a program. Refer to Appendix A, “QSYSMSG Message Queue” on page A-1 for more details. The events sent to QSYSMSG can also be logged in the audit journal. Refer to Chapter 14, “Auditing the AS/400” on page 14-1 for more details.

If QSECOFR is disabled, and no other user profile has the authority to enable it, QSECOFR can still sign on from the system console. If the console is varied off the system must be IPLed.

QRMTSIGN Remote Signon Control.

Specifies how the system handles remote signon requests.

You can find more information about how the system handles PC Support/400 remote signon requests in Chapter 12, "PC Support" on page 12-1. Also, the *AS/400: Remote Workstation Guide* contains additional information about QRMTSIGN and an example of a program to handle signon requests.

QSECURITY System Security Level.

QSECURITY controls the security level of the system. AS/400 security offers five levels of security:

- Level 10: There is no user authentication, or resource protection. No password is required to sign on. The system is shipped with this value. It should be changed immediately, preferably to 30. If you wish to move to a security level above 30, you should first test your installation on level 30. Refer to 2.2, "Security Level 40 and 50" on page 2-6 for more information.
- Level 20: Password - User authentication through user profile and password checking; no resource protection.
- Level 30: Password and Resource - User authentication and resource protection. Users require authority to access objects.
- Level 40: Password, Resource and Operating System Integrity - User authentication, resource protection, and machine interface protection.
- Level 50: Password, Resource and enhanced Operating System Integrity - User authentication, resource protection, and machine interface protection. Security level 50 is intended for AS/400 systems with high security requirements and to meet C2 security requirements.

You can find a detailed explanation about QSECURITY system values 40 and 50 later in this chapter.

2.1.2 Other Related System Values

The following system values, while not specifically security-related, affect system functions when certain security system values are set.

QAUTOVRT Automatic Configuration of Virtual Devices

Specifies whether display station passthrough virtual devices and TELNET full screen virtual devices are automatically configured. The security risks using QAUTOVRT are discussed in 11.7.2, "DSPT Virtual Configuration Descriptions" on page 11-10.

QDSCJOBITV Disconnected Job Time-Out Interval.

This system value determines if and when the system ends a disconnected job. The interval is specified in minutes.

2.1.3 System Values for Passwords

The following values apply to passwords. These values require users to change their passwords regularly as well as enforce rules for the creation of new passwords which prevents the use of passwords that are trivial or easy to guess.

Whenever you want to change any of these system values, be sure to discuss the impact with the users on the system. Do remember to inform them when any change is made.

The password composition system values are enforced only when the password is changed using the Change Password (CHGPWD) command, the ASSIST menu option to change a password, the QSYCHGPW application program interface (API), or on signon when a password expires.

In addition to the values below, passwords can be further verified by a password validation program.

QPWDEXPITV Password Expiration Interval Value.

This value forces users to change password every 1 to 366 days, or not at all. The value must be set according to the company′s security policy. This interval can also be modified for each user through the user profile password expiration interval parameter, PWDEXPITV.

QPWDLMTAJC Restrictions of Consecutive Digits in Passwords.

QPWDLMTCHR Restricted Characters for Passwords.

QPWDLMTREP Restriction of Repeated Character in Passwords.

QPWDMAXLEN Maximum Length of Passwords.

QPWDMINLEN Minimum Length of Passwords.

QPWDPOSDIF Position Difference of Characters in Successive Passwords.

QPWDRQDDGT Requirement for Numeric Characters in Passwords.

QPWDRQDDIF Required Difference in Passwords.

QPWDVLDPGM Password Validation Program.

Specifies the name of a validation program.

Both Appendix B, “Sample Password Validation Program” on page B-1 and *AS/400 Security Reference* have an example of a validation program.

2.1.4 Auditing System Values

This topic discusses the system values for controlling auditing on your system and a description of each. For more detailed information about auditing, refer to Chapter 14, “Auditing the AS/400” on page 14-1.

QAUDCTL Auditing Control.

The QAUDCTL system value determines whether auditing is performed.

QAUDENDACN Auditing End Action.

The QAUDENDACN system value determines what action the system takes if auditing is active and the system is unable to write entries to the audit journal.

QAUDFRCLVL Auditing Force Level.

The QAUDFRCLVL system value determines how often new audit journal entries are forced from memory to auxiliary storage (disk). This system value controls the amount of auditing data that may be lost if the system ends abnormally.

QAUDLVL Auditing Level.

The QAUDLVL system value determines which security-related events are logged to the security audit journal (QAUDJRN).

See Chapter 6, "Audit Journal" on page 6-1 for more information.

QCRTOBJAUD Auditing for New Objects.

The QCRTOBJAUD system value is used to determine the auditing for a new object, if the auditing default for the library of the new object is set to *SYSVAL.

2.2 Security Level 40 and 50

Security levels 40 and 50 prevent potential integrity or security risks from Machine Interface (MI) programs that could circumvent security in special cases. The level 40 controls include:

- Special object types and attributes are used and enforced at level 40 to provide OS/400 system integrity. Using them you can provide security in:
  - Preventing the use of unsupported interfaces
  - Protecting a program′s associated space
  - Protecting a job′s address space
  - Enhanced hardware storage protection
- Features that control and eventually block unauthorized use of resources that expose security and integrity:
  - Restricted instructions are prevented.
  - Validity checks have been added to the Restore Object (RSTOBJ) command.
  - Users can restrict the use of their authorities in other users′ job descriptions.
  - Password is required when using a job description with a user profile in the USER parameter in conjunction with the Add Workstation Entry (ADDWSE) command.

2.2.1 Internal System Structures Access

Figure 2-1 on page 2-7 shows the access mechanism for internal system structures.

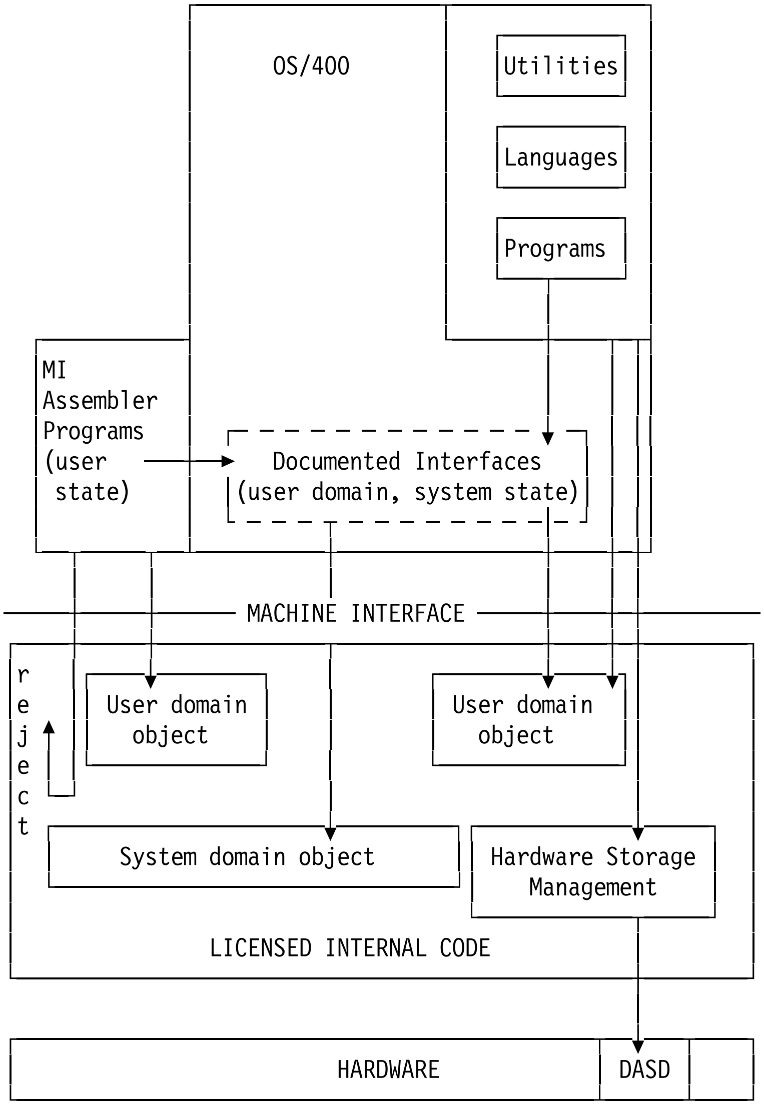


Figure 2-1. OS/400 Internal System Structure Access

Security levels 40 and 50 ensure that MI assembler programs or any other programs in user state can only access objects in user domain. Objects in system domain can be accessed only through the documented interfaces above the machine-interface level. In Figure 2-1, note the following:

- The only access to system domain objects by user state program is through the documented interfaces provided by OS/400.
- Programs in user state can directly access user domain objects such as user spaces, user indexes, or user queues.
- MI programs in user state cannot directly access system domain objects.
- Programs in user state can access blocks of information located on disk but controlled by the system. It depends on the domain of the object that contains the block of information.

2.2.2 System Object Access Control

Security levels 40 and 50 enforce a special object attribute, a *domain*, that limits program access based on the program object attribute *state*.

Special Object Attributes and Types

To prevent high-level language programs from circumventing security by direct invocation of internal system interfaces, special object attributes are provided:

DOMAIN	OS/400 object attribute
STATE	OS/400 program object attribute

Domain: Every object belongs to one of two domains, *SYSTEM or *USER. These attributes are used to isolate system objects from user objects and thereby control which programs can access the object.

The following command will display the attribute of an object:

```
DSPOBJD OBJ(library/object) OBJTYPE(objtype) DETAIL(*FULL)
```

State: Programs have an attribute, called state. The only value the attribute state can have, at execution time, is *SYSTEM, *INHERIT or *USER. *SYSTEM and *INHERIT are reserved for IBM programs.

The following command will display the state of a program:

```
DSPPGM PGM(library/program)
```

Domain/State Protection: Together, the domain and state attributes provide system object integrity. Programs in system state can access objects in both domains, while programs in user state can only access user domain objects.

Table 2-1 shows the domain/status access rules.

Table 2-1. State and Domain Access Rules at Security Levels 40 and 50

Program State	Object Domain	
	*USER	*SYSTEM
*USER	YES	NO
*SYSTEM	YES	YES

Note: If a program in *USER State tries to access a *SYSTEM Domain object, an entry is recorded in the system audit journal. At security levels 40 or 50, this operation will also fail with an MCH6801 message.

Object Types: In order to provide an object that is directly accessible in OS/400, the following object types exist:

User space (*USRSPC)	An object consisting of a collection of bytes used for storing any user-defined information.
User index (*USRIDX)	An object that provides a specific order for byte data according to the value of the data. This is similar to a key sequenced file.
User queue (*USRQ)	An object consisting of a list of messages that communicate information to other application programs.

These types of objects are in user domain to allow access by user programs. For more information about the OS/400 application program interfaces (APIs) that can be used to access data in an user space, user queue, or user index, refer to the *AS/400 Programming: System Programmer′s Interface Reference*.

Callable Application Program Interfaces

OS/400 provides application program interfaces (APIs), callable from high-level language programs that allow experienced system programmers to create applications for:

- Obtaining better performance
- Obtaining system information and functions that are not available from CL commands

These APIs are system-managed interfaces that do not violate the system′s integrity.

For a detailed explanation and examples of how to use APIs, see the *AS/400 Programming: System Programmer′s Interface Reference.*

Protecting a Program′s Associated Space

At security levels 40 and 50, a user state program cannot directly change the associated space of a program object.

Protecting a Job′s Address Space

At security levels 40 and 50, a user state program cannot obtain the address of another job in the system. Therefore, a user state program cannot directly manipulate objects associated with the other job.

Enhanced Hardware Storage Protection

Enhanced hardware storage protection allows blocks of system information located on disk to be defined as read-write, read only or no access. At security levels 40 and 50, the system controls how the *USER state programs access these protected blocks. This support is not available at security levels less than 40.

Enhanced hardware storage protection is supported on all AS/400 systems except the following:

- All B models
- All C models
- D models: 9402 D04, 9402 D06, and 9404 D20

2.2.3 Additional Level 40 and 50 Features

Other potential security and integrity exposures are prevented by level 40 and 50 features comprising:

- Preventing the use of restricted instructions
- Validation of programs being restored
- Job description authorization
- Mandatory password for signon

Validation of Programs Being Restored

A program containing restricted instructions cannot be compiled or created on an AS/400 system. The system uses a technique called *program validation* to determine whether a program being restored to your system may contain restricted instructions either because the program was created in a previous release or because the object code has been changed in the new release.

When a program is created (compiled), the system creates a validation value that is stored on the system with the program object. When the program is saved to tape, the validation value is also saved, being a part of the program object. When the program is later restored (for instance, if the program has been deleted), the validation value is recalculated, only this time from the program object stored on tape. This is then compared to the validation value of the program as stored on tape.

The program is restored only if the validation values match.

An attempt is made to retranslate the program to obtain a validation value in either of the following circumstances:

- The validation values do not match.
- The allow object differences (ALWOBJDIF) parameter value on the RSTOBJ command is *NONE.

If the translation succeeds, the translated program is restored. For programs created on V1R3 or later, an *AUTFAIL journal entry is logged in the audit journal and a message is logged in the joblog.

If the translation fails, the original program is restored. For security levels 40 or higher, and ALWOBJDIF(*NONE), all public and private authorities are revoked and the ownership is transferred to QDFTOWN. An *AUTFAIL entry is made in the audit journal and a message is logged in the job log. This entry indicates that a program that may result in a possible violation of security was restored. The *AS/400 Security Reference* has flowcharts on validation checking.

Note: Use the Force Program Recreation (FRCCRT) parameter on the Change Program (CHGPGM) command to create programs again before moving to security level 40 or higher. This enables the validation value to be calculated and prevents additional restore time.

Job Description Authorization

If a user profile name is used as a value for the User parameter in a job description, all jobs submitted with this job description are executed with attributes taken from the named user profile. This may represent a security exposure.

Security level 40 or higher requires the user to be authorized both to the job description and the user profile that is named in the job description, as described in Table 2-2 on page 2-11.

Table 2-2. OS/400 Job Description and User Profile Authorization

Object Type	Authority required by user to object	
	Level 30	Level 40 or higher
*JOBD	*USE	*USE
*USRPRF	none	*USE

Note: Attempts to use a job description containing a user profile name without authority to the user profile are logged in the QAUDJRN (the system audit function must be active). At security levels 40 or higher, this operation also fails.

Signon Without Password

At security level 30 or below, it is possible to create a subsystem description that allows users to sign on with default profiles and not enter a user profile or password. This subsystem configuration may compromise the security policies of your company. Anyone can start an interactive job using the devices allocated by the subsystem.

Security level 40 or higher blocks any attempt to sign on without a user profile and password.

At all levels, the attempts to sign on without a password are logged in QAUDJRN if the system audit function is active.

Refer to the *AS/400 Security Reference* for information about the entries in the audit journal.

2.3 Enabling Security Level 40

Make sure that all your applications run successfully at security level 30 before migrating to level 40. Security level 30 gives you the opportunity to test resource security for all your applications.

Refer to the *AS/400 Security Reference* about going from level 30 to a higher level.

2.4 Disabling Security Level 40

Installations with level 40 implemented may want to move to a lower security level in either of the following circumstances:

- New application packages are being tested for the presence of restricted MI instructions.
- Some existing applications may not have been tested for restricted MI instructions under other security levels and fail at security level 40.

Going to a lower security level allows the application programs to run while logging the violations.

To go to lower levels:

1. Use the CHGSYSVAL command to set the value of QSECURITY to the desired value.
2. Re-IPL the system.

Note: When moving from level 40 to level 30, all special authorities granted to user profiles are left unchanged. Moving to levels lower than level 30 resets the special authorities to user class defaults, including giving *ALLOBJ special authority to all users.

Hardware Storage Protection violations are not auditable below security level 40. Auditing at security level 30 no longer gives the complete picture.

2.5 Additional Level 50 Features

Security level 50 is designed to meet the requirements defined by the U.S. Department of Defense for C2 security. It provides enhanced integrity protection in addition to what is provided by security level 40. You can find more information about C2 in Chapter 8, “C2 Overview and Considerations” on page 8-1.

The following security functions are added for security level 50. They are described in the sections that follow, and in the *AS/400 Security Reference*.

- Restricting user domain object types
- Parameter Validation
- Restricting message handling between user and system state programs

2.5.1 Restricting User Domain Objects

Most object types on the system are created in system domain. When you run your system at security level 40 or 50, system domain objects can be accessed only by using the commands and APIs provided.

2.5.2 Parameter Validation

The user interfaces to the operating system are system state programs in user domain. In other words, they are programs that can be called directly by a user. Refer to Figure 2-1 on page 2-7. When parameters are passed between user state and system state programs, those parameters must be validated to ensure that an invalid parameter cannot cause the system interface to perform some unintended function or unintentionally modify a system domain object. Also, no invalid parameter value can be allowed to disrupt the normal operation of the system for other users, thus assuring its integrity.

2.5.3 Restricting Message Handling

Messages sent between programs provide the potential for integrity exposures. The following applies to message handling at security level 50:

Table 2-3. Message Handling at Security Level 50

Program Sender	Program Receiver	
	*USER state	*SYSTEM state
*USER state	any type of message	non-exception message exception message (1)
*SYSTEM state	any type of message	any type of message

Note: (1) A user state program can send an exception type message (status, notify or escape) to system state program if the system state program is a request processor or the system state program calls a user state program.

2.6 Enabling Security Level 50

Most of the additional security measures that are enforced at security level 50 do not cause audit journal entries at lower security levels. Therefore, an application cannot be tested for all possible integrity error conditions prior to changing to security level 50.

The actions that cause errors at security level 50 are uncommon in normal application software. Most software that runs successfully at security level 40 also runs at security level 50.

Refer to the *AS/400 Security Reference* for more information about changing to security level 50.

2.7 Disabling Security Level 50

After changing to security level 50, you may find you need to move back to security level 30 or 40.

Note: If you move from level 50 to level 20 or level 10, the system resets the special authorities to the user class defaults, including giving *ALLOBJ special authority to all users.

2.8 Network Attributes

Network attributes are local system values that govern how your AS/400 system participates in a network. Your AS/400 is shipped with IBM-supplied values for network attributes. To see how your network attributes are set, you can use the Display Network Attributes (DSPNETA) command. If you are going to change any of them, use the Change Network Attributes (CHGNETA) command.

From a security point of view, there are three network attributes that are important. They are:

JOBACN Job Action.

Determines how your system process incoming requests to run jobs.

DDMACC DDM Request Access.

Determines how your system process Distributed Data Management (DDM) requests from other systems, including other AS/400s, PCs, and host systems.

DDM requests can be controlled by a user-written program. You can find examples about this kind of program in Chapter 11, “Security in IBM Communications Applications” on page 11-1 and in Chapter 12, “PC Support” on page 12-1.

PCSACC PC Support Access.

The way in which the system processes PC Support/400 requests from other systems.

PC Support Access requests can be controlled by a user-written program. There are examples of this kind of program in Chapter 12, “PC Support” on page 12-1.

2.9 System Values and Network Attributes Recommendations

The following tables include brief descriptions and default values of the security-related system values and network attributes, and the recommended values if applicable to your system.

System Values for Auditing

There are five system values for auditing, listed in Table 2-4.

Table 2-4. System Values for Auditing

System Value	Description	Default Value	Recommended Value
QAUDCTL	Auditing control	*NONE	*AUDLVL, *OBJAUD
QAUDENDACN	Auditing end action	*NOTIFY	If your security policy requires that no processing be performed on the system without auditing, then you must select *PWRDWNSYS. For most installations, *NOTIFY is the recommended value.
QAUDFRCLVL	Auditing force level	*SYS	*SYS provides the best auditing performance. However, if your installation requires that no audit entries be lost when your system ends abnormally, you must specify 1. Specifying 1 may result in a noticeable performance degradation.
QAUDLVL	Auditing level	*NONE	See 6.3.4, "Auditing Level (QAUDLVL)" on page 6-3 for the recommended values.
QCRTAUDOBJ	Auditing for New Objects	*NONE	The value you select depends upon the auditing requirements of your installation. The section 6.5, "Specific User and Object Auditing" on page 6-18 provides more information about the methods for setting up the object auditing on your system.

System Values for Passwords

There are ten system values for passwords, listed in Table 2-5.

Table 2-5 (Page 1 of 2). System Values for Passwords

System Value	Description	Default Value	Recommended Value
QPWDEXPITV	Password expiration interval	*NOMAX	Consult the company′s Security Policy. Changing the password regularly limits the time period of exposure if a password is learned by someone else. **Note:** A password expiration interval can also be specified in individual user profiles.
QPWDMINLEN	Minimum length of passwords	1	5, to prevent the users from assigning the passwords that are easily guessed, such as the initials or a single character.

Table 2-5 (Page 2 of 2). System Values for Passwords

System Value	**Description**	**Default Value**	**Recommended Value**
QPWDMAXLEN	Maximum length of passwords	10	8. If the system is in a network with other systems, you should restrict the number to 8, which is normally the maximum length of password on most traditional systems.
QPWDRQDDIF	Required difference in passwords	0 (No)	If set to 1 in V2R3, the last 32 passwords may not be reused. In V3R1 the combination of QPWDEXPITV and QPWDRQDDIF determines the number of days before a password may be reused.
QPWDLMTCHR	Restricted characters for passwords	*NONE	A,E,I,O, and U for more strict security *NONE for most installations
QPWDLMTAJC	Restriction of consecutive digits for passwords	0 (No)	1 (Yes). Prevents the user from using obvious numeric sequences in a password, such as car number plate or employee serial number, which are easy to guess.
QPWDLMTREP	Restriction of repeated characters for passwords	0 (No)	1 or 2. Prevents easy to guess passwords such as “AAAA.” May allow “BANANA” but not “APPLE.”
QPWDPOSDIF	Character position difference for passwords	0 (No)	1 (Yes). Prevents changing just one character from the previous password when a user changes the password.
QPWDRQDDGT	Requirement for numeric character in passwords	0 (No)	1 (Yes). A combination of alphabetics and digits prevents the use of obvious choices such as names, brand names, popular words and so on.
QPWDVLDPGM	Password validation program	*NONE	*NONE for most installations See Appendix B, “Sample Password Validation Program” on page B-1 for a sample password validation program.

Other Security-Related System Values

There are thirteen more system values for security listed in Table 2-6.

Table 2-6 (Page 1 of 2). Other System Values for Security

System Value	Description	Default Value	Recommended Value
QALWUSRDMN	Allow user domain objects	*ALL	If your system has a high security requirements, you should allow the user domain objects only in QTEMP library. At security level 50 the QTEMP library is a temporary object and cannot be used to pass confidential data between the users. Your system may have application software that relies on *USRSPC, *USRIDX or *USRQ type objects. If so, include the libraries used by the application software on the list of libraries for the QALWUSRDMN system value. Refer to 2.5, "Additional Level 50 Features" on page 2-12.
QALWOBJRST	Allow object restore	*ALL	*NONE. This may be too restrictive for a few installations. Those who frequently install applications that adopt the owner′s authority may want to set this to *ALWPGMADP.
QAUTOVRT	Automatic configuration of virtual devices	0 (No)	0. Chapter 10, "Communications Security in SNA" on page 10-1 has more information about using display station pass-through and TELNET.
QCRTAUT	Authority for new objects	*CHANGE	*CHANGE.
QDSCJOBITV	Disconnected job timeout interval	240	120 to 240.
QDSPSGNINF	Display signon information	0 (No)	1 (Display is shown) is recommended so the users can monitor the attempted use of their profiles and know when a new password is needed. **Note:** Display signon information can also be specified in individual user profiles.
QINACTITV	Inactive job timeout interval	*NONE	30 to 60 minutes.
QINACTMSGQ	Inactive job timeout message queue	*ENDJOB	*DSCJOB unless your users run PC Support/400 jobs. Using *DSCJOB when some PC Support/400 jobs are running is the equivalent of ending the jobs. It can cause significant loss of information. Use the *message-queue* option if you have the PC Support/400 licensed program. Refer to 12.2, "Using Security System Values with PC Support" on page 12-4 which shows an example of a program to handle these messages.
QLMTDEVSSN	Limit device sessions	0 (No)	1 (Yes) because limiting the users to a single device reduces the likelihood of sharing passwords and leaving the devices unattached. **Note:** Limiting device sessions can also be specified in individual user profiles.

Table 2-6 (Page 2 of 2). Other System Values for Security

System Value	Description	Default Value	Recommended Value
QLMTSECOFR	Limit security officer	1	1 (Yes).
QMAXSIGN	Maximim signon attempts	15	3. The number should be high enough to allow possible typing errors but not enough to allow attempts to guess the password.
QMAXSGNACN	Action when signon attempts reached	3	3 (Disable both the user profile and device). Refer to 12.2, “Using Security System Values with PC Support” on page 12-4 for more information of this system value and its use with PC Support.
QRMTSIGN	Remote signon control	*FRCSIGNON	*REJECT if you do not want to allow any pass-through or PC Support/400 access. If you do allow pass-through or PC Support/400 access, using a pass-through validation program provides the best control. Appendix E, “Example Exit Program for QRMTSIGN System Value” on page E-1 contains a program for controlling remote signon attempts.
QSECURITY	System security level	10	40. See 2.2, “Security Level 40 and 50” on page 2-6 and 2.5, “Additional Level 50 Features” on page 2-12 for more information about 40 and 50 security levels.

Security-Related Network Attributes Values

The following table shows the network attributes that apply to security. As in the previous sections, you will find the network attribute, a brief description and the recommended value.

Table 2-7. Security-Related Network Attributes

Network Attribute	Description	Default Value	Recommended Value
JOBACN	Job action	*FILE	If you do not expect to receive remote job requests on your system set the JOBACN network attribute to *REJECT.
PCSACC	PC Support access	*OBJAUT	*name-of-a-program*. You can find more information in Chapter 12, “PC Support” on page 12-1.
DDMACC	PC Support access	*OBJAUT	*name-of-a-program*. You can find more information in Chapter 11, “Security in IBM Communications Applications” on page 11-1 and an example of such a program in Appendix D, “Program Used with DDMACC on Network Attributes” on page D-1.

Chapter 3. User Profiles and Group Profiles

The user profile on the AS/400 system is a basic, but powerful control mechanism. It controls what the user can do and customizes the way the system appears to the user.

A group profile is a special type of user profile. You can use a group profile to define authority for a group of users, rather than giving authority to each user individually. A group profile can own objects on the system.

This chapter discusses more about user profiles and group profiles. For detailed information see *AS/400 Security Reference*.

3.1 Individual User Profiles

User profiles contain security-related information, such as:

- How the user signs on the system
- What the user is allowed to do after signing on
- What objects are owned by the user
- What private authorities the user has to the objects
- How the user′s actions are audited.

If the security level (system value QSECURITY) on the system is 10, the system automatically creates a user profile when someone attempts to sign on with a user profile that does not exist on the system. If the security level on the system is 20 or higher, a user profile *must* exist before a user can sign on.

You must have *SECADM special authority in your user profile and be authorized to the user profile to create, change, and delete user profiles.

You can create user profiles in several ways:

- From the Work with User Profiles (WRKUSRPRF) display
- Using the Create User Profile (CRTUSRPRF) command
- Using the Work with User Enrollment option from the SETUP menu
- From the Enroll Office User display within OfficeVision/400

You cannot delete a user profile that owns objects or is the primary group for an object. You must delete any objects owned by the user profile or transfer ownership of those objects to another profile. However, you can delete a user profile even though that profile still owns spooled files. After you have deleted a user profile, use the Work with Spooled Files (WRKSPLF) command to locate and delete any spooled files owned by the user profile. When you delete a user profile, the user is removed from all distribution lists and from the system directory. You do not need to change ownership of or delete the user′s message queue. The system automatically deletes the message queue when the profile is deleted.

If a user is enrolled in OfficeVision/400, you need to remove that user from the system using the OfficeVision/400 administration function. This also provides an option to remove the user from the system. There is one facility that can remove an Office user outside of Office administration. Through Work with User Profile (WRKUSRPRF) command, you can delete a user profile and delete owned objects, transfer them to another user, or a combination of both. This interface will also remove the user from Office before deleting the profile.

As well as being able to create, change and delete user profiles, you can also copy, list, enable or disable user profiles.

3.1.1 Password

A user′s password is an important key to controlling access to system resources. If a password is disclosed, the system integrity is reduced. As such, users should be educated about the need for good practices in password control and management.

Password management involves enforcement of several rules:

- Password must be changed at certain intervals.
- The same password cannot be reused.
- Non-trivial words of a reasonable length must be used.
- Users must keep passwords secret.

Password management is facilitated through the use of system values and user profile parameters. User education is an important component of password management that must be done by system administrators.

Password Change Required Within Certain Intervals

The password expiration interval, system value QPWDEXPITV, can force users to change passwords every one to 366 days, or not at all. The company′s security policy dictates the interval. This interval can be also be modified for each user through the user profile password expiration interval parameter, PWDEXPITV. The IBM supplied default for QPWDEXPITV is *NOMAX, indicating that no limit is enforced.

Prevent Reuse of the Same Password

Users change their passwords with the change password (CHGPWD) command. This command prompts the user for the old password, new password, and a re-verification of the new password. The new password may not be the same as the old password.

QPWDRQDDIF determines if the new password must be different from previous passwords. If the setting of QPWDRQDDIF is different from 0:

- In V2R3 and earlier a user must select a password that is different from the 32 previous passwords.
- In V3R1 more values have been provided for QPWDRQDDIF to offer more granularity in determining how often a password can be reused.

The IBM-supplied default for QPWDRQDDIF is ′0′.

Use of Non-trivial Words of a Reasonable Length

When a user is initially registered into the system, or when a user forgets his or her password, the security officer may initialize the password to match the user profile, by specifying *USRPRF in the Password field. However, to force the user to change the password to something less obvious, the PWDEXP (set password to expired) user profile field should be set to *YES, which forces the user to change passwords at the next signon.

In addition, the system values shown in Table 2-5 on page 2-14 can be used to prevent the use of easy-to-guess passwords.

System restrictions on password make-up are only applicable when the password is changed with the change password (CHGPWD) command, the API, or during signon, if the password has expired. Rules do not apply when using Create User Profile (CRTUSRPRF) or Change User Profile (CHGUSRPRF) commands. Because of this, a user′s password should always be set to expired when it is changed with the CRTUSRPRF or CHGUSRPRF commands.

3.1.2 Special Authority

Special authority is used to specify the certain actions a user can perform on system resources and tasks. A user can be given one or more special authorities.

*ALLOBJ Special Authority

All object (*ALLOBJ) special authority allows the user to access any resource on the system whether or not private authority exists for the user. Even if the user has *EXCLUDE authority for an object, *ALLOBJ special authority still allows the user to access the object.

Only a very limited number of user profiles should be given *ALLOBJ special authority, and you may want to audit these users.

If a user requires *ALLOBJ special authority to perform a specific task, consider using a program that adopts its owners authority instead.

*SECADM Special Authority

Security administrator (*SECADM) special authority allows a user to create, change, and delete user profiles. In addition, *SECADM special authority gives the user comprehensive authority to manage OfficeVision/400 objects and users.

An OfficeVision administrator may be given full or limited *SECADM special authorities. An administrator with limited *SECADM special authorities cannot work with system objects while using the OfficeVision/400 program.

Only a very limited number of user profiles should be given *SECADM special authority, and you may want to audit these users.

*JOBCTL Special Authority

Job control (*JOBCTL) special authority gives the user control over jobs running in the system. It is a powerful authority, and only a limited number of users should be given *JOBCTL special authority. You may want to audit these users.

*SPLCTL Special Authority

Spool control (*SPLCTL) special authority gives the user control over all output queues and job queues in the system.

It is a powerful authority and only a very limited number of user profiles should be given *SPLCTL special authority. You may want to audit these users.

4.9.1, “Security in Output Queues” on page 4-9 has a detailed discussion of authority exposures of users with *SPLCTL special authority.

*SAVSYS Special Authority

Save system (*SAVSYS) special authority gives the user the authority to save, restore, and free storage for all objects on the system, whether or not the user has object existence authority to the objects.

Only a limited number of users should be given *SAVSYS special authority.

If you are concerned about industrial espionage (and you should be), here is a suggestion on how to further restrict copying to tape/diskette:

- Change the public authority to *EXCLUDE on all the SAVxxx and RSTxxx commands.
- Change the public authority to *EXCLUDE on tape and diskette device descriptions.
- Get control over who can use PC Support transfer to copy data to and from a PC or PS/2. Refer to Chapter 12, “PC Support” on page 12-1 for more information.
- Verify if anyone can use DDM to copy a file. Refer to 11.5.4, “DDM Access Parameter (DDMACC)” on page 11-7 for more information.
- To perform periodic backups create a user profile with:
  - SPCAUT = *SAVSYS.
  - LMTCPB = *YES.
  - INLPGM = a program containing the save commands. The program must adopt its owner′s *SAVSYS authority.
  - INLMNU = *SIGNOFF.

  When an operator signs on with this user ID the program is executed, and at the end of the job the sign on screen is presented.

*SERVICE Special Authority

Service (*SERVICE) special authority allows the user to perform the display and alter service functions. The dump trace and dump tape commands can be performed without *SERVICE authority. See 3.7, “*SERVICE Users Considerations when Using Communication Line Trace” on page 3-11 for more information about obtaining service trace data.

Only a limited number of users should be given *SERVICE special authority.

*AUDIT Special Authority

Audit (*AUDIT) special authority gives the user the ability to change auditing characteristics. This authority should only be given to users who are responsible for auditing the security.

*IOSYSCFG Special Authority

System configuration (*IOSYSCFG) special authority gives the user the ability to change how the system is configured. *IOSYSCFG is only required to use configuration commands that are new in V3R1, for example Add Community SNMP (ADDCOMSNMP).

3.1.3 User Class

User class is used to control what menu options are shown to the user on OS/400 menus, and also offers a convenient way of specifying special authorities.

Table 3-1 shows the possible user classes and the default special authorities for each user class. The entries indicate that the authority is given at security levels 10 and 20 only, at all security levels, or not at all.

Table 3-1. Default Special Authorities by User Class

Special Authority	**User Classes**				
	***SECOFR**	***SECADM**	***PGMR**	***SYSOPR**	***USER**
*ALLOBJ	All	10 or 20	10 or 20	10 or 20	10 or 20
*SECADM	All	All			
*JOBCTL	All	All	All	All	
*SPLCTL	All				
*SAVSYS	All	All	All	All	10 or 20
*SERVICE	All				
*AUDIT	All				
*IOSYSCFG	All				

When creating a new user profile with the CRTUSRPRF command, the default user class is *USER, which in a level 30 or above system means the user gets no special authorities.

3.1.4 Initial Program

You can specify the name of a program to call when a user signs on.

3.1.5 Initial Menu

You can specify the name of a menu to be shown when the user signs on. The initial menu is displayed after the user′s initial program runs.

If you want the user to run only the initial program, you can specify *SIGNOFF for the initial menu. For example, if you want the end users to process only the menu options created by the initial program, and no opportunity to enter any commands from command entry, you can specify *SIGNOFF for the initial menu, and when the end users exit the menu (the initial program is over), they will sign off automatically, to prevent them having any opportunity to execute commands.

3.1.6 Limited Capabilities

Limit Capabilities is a commonly used, simple, but powerful facility for controlling user access to the system and its resources. You can use the Limit Capabilities field in the user profile to limit the user′s ability to enter commands and to override the initial program, initial menu, current library, and attention program specified in the user profile.

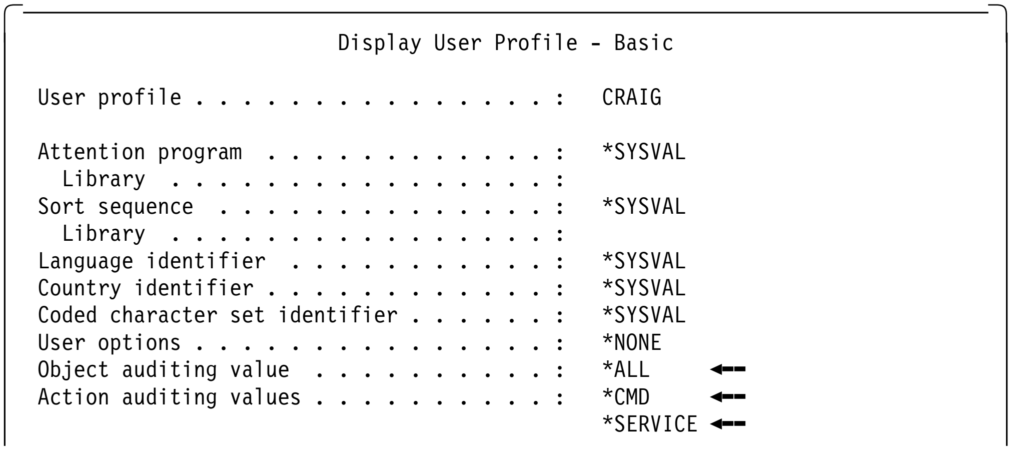
3.2 User Auditing

For V2R3 and later releases, you can define auditing on your system at three different levels:

- System-wide auditing that occurs for all users
- Auditing that occurs for specific objects (object auditing)
- Auditing that occurs for specific users (action auditing)

If you have a sensitive data file, for example, SALARY, and you want to know who has accessed it, and what actions the user has taken on it, you can define object auditing for SALARY. If you have a privileged user profile, for example, user profile CRAIG with *ALLOBJ special authority, and you want to know CRAIG′s every action in your system, you can define action auditing for user profile CRAIG. Or if you only want to know CRAIG′s actions on file SALARY, you still can achieve this. You need to use a combination of system values, user profile parameters, and object parameters to define auditing. Chapter 6, “Audit Journal” on page 6-1 will describe how to do this. In this topic, we only focus on the discussion of user profile parameters.

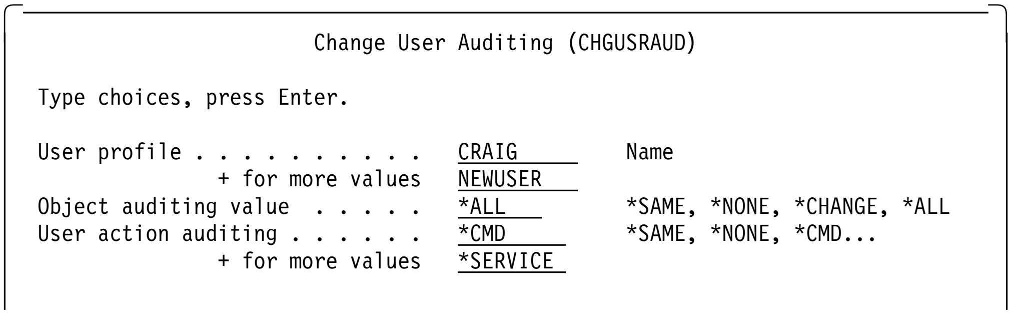
You can use the Display User Profile (DSPUSRPRF) command to see audit characteristics for a user. As Figure 3-1 shows, there are two parameters, Object auditing value and Action auditing values, which relate to user-based auditing.



```
                         Display User Profile - Basic

 User profile . . . . . . . . . . . . . . . :   CRAIG

 Attention program  . . . . . . . . . . . . :   *SYSVAL
   Library  . . . . . . . . . . . . . . . . :
 Sort sequence  . . . . . . . . . . . . . . :   *SYSVAL
   Library  . . . . . . . . . . . . . . . . :
 Language identifier  . . . . . . . . . . . :   *SYSVAL
 Country identifier . . . . . . . . . . . . :   *SYSVAL
 Coded character set identifier . . . . . . :   *SYSVAL
 User options . . . . . . . . . . . . . . . :   *NONE
 Object auditing value  . . . . . . . . . . :   *ALL     ◄━
 Action auditing values . . . . . . . . . . :   *CMD     ◄━
                                                *SERVICE ◄━
```

Figure 3-1. Display of the DSPUSRPRF Command

Although you can see Object and Action auditing values from the Display User Profile (DSPUSRPRF) command, you can′t set them or change them with any user profile commands, for example, the Create User Profile (CRTUSRPRF) command, the Change User Profile (CHGUSRPRF) command, and so on. You have to use the Change User Auditing (CHGUSRAUD) command to set the audit characteristics for users. Only a user with *AUDIT special authority can use the CHGUSRAUD command.



```
                         Change User Auditing (CHGUSRAUD)

 Type choices, press Enter.

 User profile . . . . . . . . . .   CRAIG         Name
               + for more values    NEWUSER
 Object auditing value  . . . . .   *ALL          *SAME, *NONE, *CHANGE, *ALL
 User action auditing . . . . . .   *CMD          *SAME, *NONE, *CMD...
               + for more values    *SERVICE
```

Figure 3-2. Display of the CHGUSRAUD Command

As Figure 3-2 shows, you can specify the auditing characteristics for more than one user at a time by listing user profile names. The AUDLVL (user action auditing) parameter can have more than one value.

3.2.1 Object Auditing

The object auditing (OBJAUD) value for a user profile works with the object auditing value for an object to determine whether the user′s access of the object is audited.

3.2.2 Action Auditing

For an individual user, you can specify which security-related actions should be recorded in the audit journal. The actions specified for an individual user apply in addition to the actions specified for all users by the QAUDLVL system value. The AUDLVL (user action auditing) parameter in the CHGUSRAUD can have more than one value. The values you specify in the system value override the current AUDLVL values for the users you specify with the CHGUSRAUD command.

See Chapter 6, “Audit Journal” on page 6-1 and the *AS/400 Security Reference* for more information about object auditing, action auditing, and related planning.

3.3 IBM-Supplied User Profiles

A number of user profiles are shipped with OS/400. These IBM-supplied user profiles are used as object owners for various system functions. Some system functions also run under specific IBM-supplied user profiles. In V2R3 and earlier a few of them have passwords, and these passwords must be changed. In V3R1 only QSECOFR has a password, and it must be changed. All others are set to a password of *NONE, however, only QSYSOPR, QPGMR, QUSER, QSRV and QSRVBAS can be changed.

Note: If installing V3R1 over an existing release, the passwords for these user profiles are not changed. The passwords for QSYSOPR, QPGMR, QUSER, QSRV and QSRVBAS are only set to *NONE when scratch-installing an AS/400 system.

3.3.1 Changing Passwords for Dedicated Service Tools (DST)

Dedicated Service Tools (DST) is a group of service functions used to service the system and manage disk devices when the operating system is not running. One of the important functions that DST offers is the ability to reset the QSECOFR password. DST activities are extremely sensitive and use should be severely limited. You need the DST password to access DST functions. To enter the DST environment, the keylock switch must be in the manual position, and an attended IPL of the system be performed.

There are three security levels within the DST environment. These can be seen as being three user profiles, but no user profiles exist as such. Access to the functions each level provides depends on the user entering the appropriate password for that level. The levels and their default passwords are shown in Table 3-2.

Table 3-2. Security Levels of Dedicated Service Tools

DST Security Level	**Default Password**	**User**
Security	QSECOFR	Used by a user to perform all DST functions, including changing the DST passwords.
Full	22222222	Used by a service representative or an experienced system user to provide access to all DST functions except changing the DST passwords.
Basic	11111111	Used by a service representative or an experienced system operator to provide access to functions that do not access sensitive data. See the publication *Backup and Recovery - Advanced* for more information.

Default DST passwords are the same for every AS/400 system that is shipped and should be changed to protect the security of your system. You cannot change DST passwords using the CHGUSRPRF command. QSECOFR can use the CHGDSTPWD command to reset the Security password to QSECOFR. The passwords for Full and Basic can only be changed through the DST function.

Operating System Install Security

The DST function can be used to secure the installation of the operating system.

3.3.2 Default Owner (QDFTOWN) User Profile

The Default Owner (QDFTOWN) user profile is an IBM-supplied user profile that is used when an object has no owner or when object ownership might pose a security exposure. Following are situations that cause ownership of an object to be assigned to the QDFTOWN profile:

- If a profile which owns objects becomes damaged and is deleted, its objects no longer have an owner. Using the Reclaim Storage (RCLSTG) command assigns ownership of these objects to the default owner (QDFTOWN) user profile.
- If an object is restored and the user profile which was the owner at the time of saving does not exist, then the object owner is set to QDFTOWN.
- If a program that needs to be created again (as determined by the program validation value, see “Validation of Programs Being Restored” on page 2-10)

is restored, but the program creation is not successful, then the program owner is set to QDFTOWN.

- If the maximum storage limit is exceeded for the user profile that owns an authority holder for a file being moved, renamed, or whose library is being renamed, then the object owner is set to QDFTOWN.

The system supplies the QDFTOWN user profile because all objects must have an owner. By default, only a user with *ALLOBJ special authority can display and access this user profile and transfer ownership of objects associated with the QDFTOWN user profile. You can grant other users authority to the QDFTOWN profile. QDFTOWN is shipped with password of *NONE to prevent signon.

The command

```
DSPUSRPRF USRPRF(QDTFOWN) TYPE(*OBJOWN)
```

shows the objects owned by QDFTOWN. It tells a lot about how ownership is handled on an AS/400.

3.4 Group Profiles

A group profile is a fundamental security structure used on most AS/400 systems, and is especially useful when several users have similar security requirements. If members of a department require access to the same applications, a group profile can be set up for the department. As users join or leave the department, the group profile field in their user profiles can be changed. This is easier to manage than adding or removing individual user access to objects.

In V3R1 a user may be a member of up to 16 group profiles. The first group profile is specified in the GRPPRF parameter of the user profile. Additional groups are specified in the SUPGRPPRF parameter. In earlier releases a user can only be a member of one group.

Authority may be given for group profiles to use certain objects on the system. A member of the group gets the group′s authority unless an authority is specifically defined for that user.

You should create profiles specifically to be group profiles. Be sure to specify PASSWORD(*NONE). A group profile is simply a user profile. It becomes a group profile when another profile designates *it* as the group profile. For example:

1. Create a profile called GRPSALES:

   ```
   CRTUSRPRF USRPRF(GRPSALES) PASSWORD(*NONE)
   ```

2. When the profile is created, it is an ordinary profile, not a group profile.
3. Designate GRPSALES as the group profile for another group profile:

   ```
   CHGUSRPRF CRAIG GRPPRF(GRPSALES)
   ```

4. The system now treats GRPSALES as a group profile.

If you use the EDTOBJAUT command on a user profile you can see if it is used as a group profile. In a group profile every user profile in the group gets USER DEF (all except *OBJEXIST, *OBJALTER, *OBJREF, and *EXECUTE) authority to the group profile.

3.4.1 Group Ownership of Objects

When an object is created, the system looks at the profile of the user creating the object to determine object ownership. If the user is a member of a group profile, the OWNER field in the user profile specifies whether the user or the group should own the new object.

If the group owns the object (OWNER is *GRPPRF), the user creating the object is not automatically given any specific authority to the object. The user gets authority to the object through the group. If the user owns the object (OWNER is *USRPRF), the group′s authority to the object is determined by the GRPAUT field and the GRPAUTTYP field in the user profile. The group authority becomes a private authority or a primary group authority to the object. If the user who owns the object changes to a different user group, the original group profile still retains authority to any objects created.

If you want to secure an application on an AS/400 you must have complete control of the ownership of the objects. The ideal situation is when the same user profile owns all the libraries within an application, and all the objects within each library.

The owner will decide which authority the other users can have, through a group profile and as an individual.

If you give the ownership to an application in the production environment to a group profile, you will have a serious security exposure. It may be beneficial to have group ownership in a test and development environment. At the time the objects are moved to the production environment ownership must be transferred to the application owner.

3.4.2 Planning Group Profiles

Creating profiles specifically to be group profiles is preferable to making existing profiles into group profiles. You may find that a specific user has all the authorities needed by a group of users and be tempted to make that user profile into a group profile. However, using an individual′s profile as a group profile may cause problems in the future:

- If the user whose profile is used as the group profile changes responsibilities, a new profile needs to be designated as the group profile, authorities need to be changed, and object ownership needs to be transferred.
- All members of the group automatically have authority to any objects created by the group profile. The user whose profile is the group profile loses the ability to have private objects, unless that user specifically excludes other users.

Try to plan group profiles in advance. Create specific group profiles with password *NONE. If you discover after an application has been running that a user has authorities that should belong to a group of users, do the following:

1. Create a group profile.
2. Use the GRTUSRAUT (Grant User Authority) command to give the user's authorities to the group profile.
3. Remove the private authorities from the user, because they are no longer needed. Use the RVKOBJAUT or EDTOBJAUT command.

3.5 Group Profiles and Performance

In V3R1 a user may be part of up to 16 group profiles. In earlier releases a user can only be part of one group.

Group profiles are widely used by many AS/400 installations because they are a convenient method to manage security. The management of security is easier, and also the number of authorizations stored by the system reduced which requires less time to backup the system.

A primary group may be specified for an object. This primary group is stored within the object as shown in Figure 1-1 on page 1-2. Specifying a primary group for an object may reduce the system resources used to verify the user′s authority to the object. However, the additional use of private authorities and authorization lists influence authority verification.

The *AS/400 Security Reference* has flowcharts and case studies which illustrates the sequence of the authority checking process.

Security should be implemented in such a way that it is easy to understand and maintain. Performance should be considered, but not made a main issue. See Chapter 7, “Security Design For Performance” on page 7-1 for more information about security performance.

3.6 Restoring User Profiles Considerations

If you are performing a recovery of a system, and are restoring user profiles, OS/400 attempts to maintain integrity by preventing possible security breaches.

After the user profiles are restored, the passwords from the time of the save are in effect. Remember to change the QSECOFR password as soon as a command line is available.

3.7 *SERVICE Users Considerations when Using Communication Line Trace

Sometimes program debugging or network management tasks require a way to see the data sent and received on a communications line. For this, the AS/400 provides you with communication line trace facilities. You can obtain a communication line trace in several ways. You can use the Start System Service Tools (STRSST) command menu interface, or you can use the commands listed below:

STRCMNTRC Start a communications trace for a specified line.

ENDCMNTRC Ends the trace currently running on the specified line.

PRTCMNTRC Writes the communications trace data for the specified line to a spool file of a database file.

You must be aware that if the user is able to start and print a communications line trace, **the user is able to read everything that is sent and received in that line, including the user profile names and passwords**.

Note these communications trace commands are shipped with public authority *EXCLUDE, and *SERVICE special authority is needed to use them. We recommend that you be careful when granting authority to these types of tasks and, if you grant it, keep track of the users who have it, and periodically review their operations.

These commands are good candidates for object-level auditing. See Chapter 6, “Audit Journal” on page 6-1 for more information on how to set up object auditing on these commands.

Chapter 4. Resource Protection

Resource protection defines which users are allowed to use which objects on your system, and what operations they are allowed to perform over those objects. In this chapter, the different approaches for resource protection and related management issues are discussed with recommendations for implementation. Also highlighted are considerations about users with special authorities and the security exposures they can create. Additionally, security of printed output is also considered.

4.1 Who Can Access Information

To access the resources on your system, a user must be authorized to manage these resources. You can give authority to access objects to individual users, specific groups of users, primary groups, and any other users (known as Public).

Private Authority You can define specific authority to an object. You can grant authority to an individual user or to a group profile. An object has *private* authority if any other authority than public is defined for that object. In this way, an object may have private authority for a user(s) or a group. A member of the group gets their authority from the group unless the user has a private authority.

Primary Group You can define a primary group for an object, and the authority this group has to the object. The primary group name is stored with the object and may help simplify authorization management. It may also improve authority checking performance.

Public Authority The authority given to users who do not have any specific (private) authority to an object, who are not on an authorization list (if one is specified for the object), and whose group profile has no specific authority to the object. The system uses public authority when there is no other authority specified for that user profile on the object.

Object Ownership Every object in the system has an owner. The owner has *ALL authority to the object, but this authority can be changed or revoked.

4.2 How a User Can Access Information

To access any object in the system, you must have the appropriate *authority* to that object. Authority determines the type of access you are allowed to the object. Authority to an object is divided into two categories:

Object Authority Defines which operations can be performed on the object as a whole.

Data Authority Defines operations that can be performed on the contents of the object.

In the following tables, you can find the different types of authorities that OS/400 implements.

Table 4-1. Object Authority Types

Authority Name	**Descriptive Name**	**Functions Allowed**
*OBJOPR	Object Operational	Display the description of an object. Use the object as determined by the user′s data authorities.
*OBJMGT	Object Management	Specify the security for the object. Move or rename the object. Add members to database files.
*OBJEXIST	Object Existence	Delete the object. Free storage of the object. Perform save and restore operations for the object[1]. Transfer ownership of the object.
*OBJALTER	Object Alter	Alter the attributes of database files. Change the attributes of SQL packages.
*OBJREF	Object Reference	Specify a database file as the first level in a referential constraint.
*AUTLMGT	Authorization List Management	Add and remove users and their authorities from the authorization list[2].
Note: (1) If a user has save system (*SAVSYS) special authority, object existence authority is not required to perform save and restore operations on the object. See 4.10, "*SAVSYS Special Authority Considerations" on page 4-14. (2) See Chapter 5, "Authorization Lists" on page 5-1 for more information.		

Table 4-2. Data Authority Types

Authority Name	**Descriptive Name**	**Functions Allowed**
*READ	Read	Display the contents of the object, such as viewing records in a file.
*ADD	Add	Add entries to an object such as adding jobs to a job queue or adding records to a file.
*UPD	Update	Change the entries in an object, such as changing records in a file.
*DLT	Delete	Remove entries from an object, such as removing messages from a message queue or deleting records from a file.
*EXECUTE	Execute	Run a program, service program, or SQL package. Locate an object in a library or a directory.

4.2.1 Commonly Used Authorities

The system defines a commonly used set of authorities to let you perform operations on the objects and its data. They are *ALL, *USE, *CHANGE and *EXCLUDE. *EXCLUDE authority is different than having no authority. *EXCLUDE authority specifically denies access to the object, while having no authority means that you use the public authority defined for the object.

The following table shows the types of authority and the equivalent commonly used authority name.

Table 4-3. Authority Combinations

	***OBJOPR**	***OBJMGT**	***OBJEXIST**	***OBJALTER**	**OBJREF**	***READ**	***ADD**	***UPD**	***DLT**	***EXECUTE**
*ALL	x	x	x	x	x	x	x	x	x	x
*CHANGE	x					x	x	x	x	x
*USE	x					x				x
*EXCLUDE										

Note: The common authority descriptions (left column) are equivalent to the basic authorities shown (top row). For example, specifying *USE is exactly the same as specifying *OPR, *READ and *EXECUTE. *EXCLUDE is not the absence of authority; it is an authority that prevents access to the object.

If one of the common authorities (*ALL, *CHANGE, *USE, *EXCLUDE) is not appropriate, some combination of the basic elements can be specified. For example, a certain application might require that records in a transaction log cannot be read, updated, or deleted. It should only be possible to add records. In this case *OBJOPR and *ADD would be specified, since none of the common authorities exactly matches the requirement. Another example of this would be a message queue; *OBJOPR and *ADD allow sending a message to the queue but not viewing the messages already in the queue. This is shown as USER DEF on the Display Object Authority (DSPOBJAUT) or Edit Object Authority (EDTOBJAUT) commands.

4.3 What Information Can Be Accessed

You can define resource security for both individual or groups of objects in the system. To secure groups of objects you can use library security or authorization lists.

4.3.1 Library Security

All objects in the system reside in libraries. To access an object you need authority to both the library and the object itself. When you want to secure an application, you should start with securing the libraries:

- Find out which libraries are included in the application. Don′t forget the libraries with the source code.
- Find out which user profiles must have access to the libraries, and if any of them needs specific authorities.
- At this point in time you have to decide whether to use group profiles, primary group authority, authorization lists, or a combination. Using primary group authority without authorization lists makes authorization checking faster. See the flowcharts in the *AS/400 Security Reference*. Plan to secure the objects with proper access control and the best maintainability as the primary target, but have performance in mind.

When this works at it should you can restrict the access to single objects within the libraries if that is a requirement.

4.3.2 Authorization List Security

You can group objects with similar security requirements using an authorization list. Each user can have a different authority to the set of objects secured by the list.

For more information about authorization lists, refer to Chapter 5, “Authorization Lists” on page 5-1 and the *AS/400 Security Reference*.

Authorization Lists Compared to Group Profiles

Group profiles and authorization lists provide overlapping functions, but the distinction is illustrated in Figure 4-1.

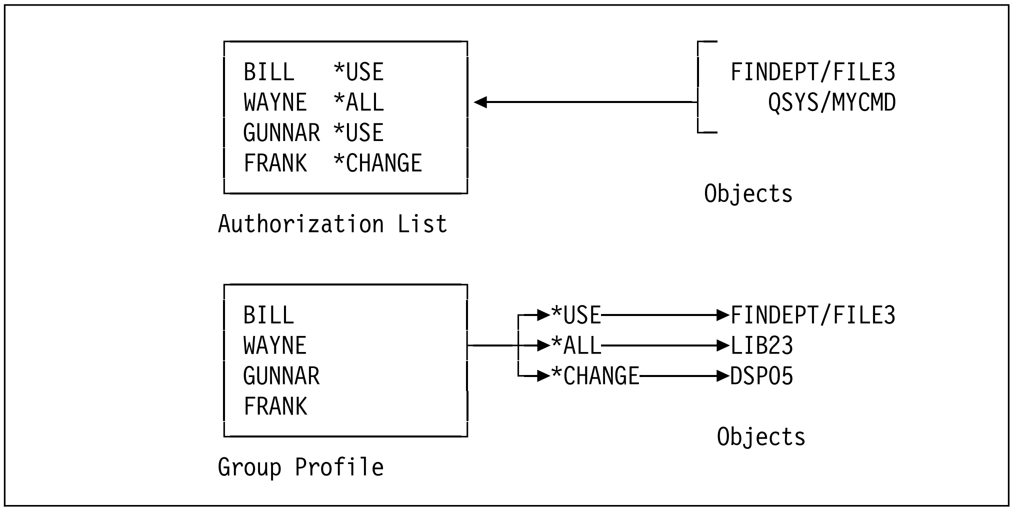


Figure 4-1. Authorization List versus Group Profile. The difference between group profile and authorization list authority handling is that they are inverse to each other.

In an authorization list, each user may have a different level of authority. Whatever that authority level is, it applies to all objects accessed through the list. If WAYNE has *ALL authority (to the authorization list) he has *ALL authority to every object secured by the list. In a Group Profile, every user has the same level of authority to a particular object. If the Group Profile has *CHANGE authority to DSP05 then every user in the group has this authority. For many people, the Group Profile method is slightly more intuitive. Refer to 5.10, “Comparing Authorization Lists to Group Profiles” on page 5-11 for more information.

4.4 Authority for New Objects in a Library

Every library has a parameter called CRTAUT (create authority). This parameter determines the default public authority for any new object that is created in that library.

If the QCRTAUT system value or the CRTAUT parameter on a library is changed *after* objects have been created in the library, this does not affect the public authority of the objects secured in this manner. This only takes effect at the create time of the object.

Refer to the *AS/400 Security Reference* for more information.

4.4.1 Create Authority (CRTAUT) Risks

If your applications use default authority for new objects created during application processing, you should control who has authority to change the library descriptions. Changing the CRTAUT authority for an application library could allow unauthorized access to new objects created in the library.

4.5 Object Ownership

Each object is assigned an owner when it is created. The owner is either the user who creates the object, or the group profile if the member user profile has specified that the group profile should be the owner of the object. Refer to 3.4.1, “Group Ownership of Objects” on page 3-10 for more information about group ownership, and to the *AS/400 Security Reference* for more information about object ownership.

4.5.1 Adopted Authority

Adopted Authority is a very important security facility of the AS/400. Normally, programs and commands operate at the authority level of the user invoking them. Thus, the same program might be executed at different authority levels, depending on who is using it. This is not always convenient, and the AS/400 provides an alternative.

When a program is created, it is possible for the owner to specify ″adopted authority″. This means the program will ″adopt″ the authority of the *owner* whenever it executes, in addition to the authority of the user. When the program ends, the additional authority is revoked. In this way, adopted authority can be considered temporary authority.

The advantage of using adopted authority is that no direct authority to objects (such as files) need be given to users of an application program, yet the users can access the file through the program. In this case the program would execute with the adopted authority of the program′s owner. (This assumes, of course, that the program′s owner has higher authority (or specific authority to the files) than normal users). Thus access to the files will be only through specific user programs that can enforce any particular data or data field security that is appropriate. This is a way of establishing and maintaining application integrity; only programs that have been properly designed, reviewed, and tested are allowed to manipulate data. This can be done without giving the user of the programs any direct access to the data involved.

Potential Exposures

Adopted authority has good and bad aspects. When properly controlled, it adds a very convenient and powerful function to the system with no loss of security. However, it can be difficult to inspect and control every program that uses adopted authority.

Particular exposures occur when:

- Someone with a higher authority, like *ALLOBJ, is the owner of local programs.
- The program calls other programs via the CALL command/statement.
- The program puts up a command line.
- Unqualified program calls (without the library name) are made, thus OS/400 searches the library list (*LIBL) for the called program. This allows for the introduction of (unauthorized) code that is executed instead of the intended program.

In designing applications that use adopted authority, you should ensure that these exposures do not arise. Authorization should be reviewed for libraries on the system, or the user′s ability to manipulate his library list. The COBOL/400 programming languages does not allow qualified program calls. RPG/400, C/400 and CL do allow qualification.

The DSPPGMADP command shows which user profile owns programs that adopt the owner′s authority. The primary profiles to check are those with special authorities.

4.5.2 Authority Holder

Another security object is the *authority holder*. The purpose is to allow an application to delete and recreate a file without losing the various authorities associated with it. The authority holder′s use will almost always be with an existing S/36 application. It is an object which contains no data. When an authority holder is created the authority from the file is transferred to the authority holder. If a program-defined database file is created with the same name as the authority holder, all the authorities associated with the authority holder are automatically transferred to the file. If the file is deleted, the authorities are automatically moved back to the authority holder.

Authority Holder Risks

Refer to the *AS/400 Security Reference* about the potential risks involved with authority holders.

4.6 *PUBLIC Authority

Use of public authority can greatly reduce the need for specific authorization (individual or group) or the use of authorization lists. If a file, for example, is not confidential, the public authority should be *USE. *USE allows anyone to read it, without needing any more security controls. If everyone in the organization needs to read the file, this is much better than making the public authority *EXCLUDE and giving specific read authorization to anyone who asks for it. The objective is to avoid unnecessarily large amounts of private authorities.

We recommend that the public authority of objects be considered as carefully as all other security aspects. The default public authority for most objects is *CHANGE.

A common (and good) protection scheme is to provide a reasonable public authority for almost all objects in a library. (Only more sensitive objects would have public authority *EXCLUDE.) The library itself (since the library is also an object) would provide the basic level of control. Thus very few of the objects in the library would require specific authorizations. With a single authorization (to the library), a user gains access to almost everything in the library. This is the basis of ″library authorization″. It is an effective way to reduce the total number of authorities in the system; this improves system performance -- especially for saves and restores.

4.7 Protection Strategies

There are three different strategies for implementing security on the AS/400 system:

- Library Security
- Object Security
- Menu Security

These different approaches can be used by themselves, or combined, and are outlined below.

4.7.1 Library Security

Library security establishes security at the library level. This concept assumes that libraries contain objects with similar protection requirements and that, in general, nonspecific protection is adequate. This concept typically applies when applications are maintained in separate libraries, and test and production objects are separated at the library level.

4.7.2 Object Security

Object security defines authorization at the more granular object level, that is below the library level. It is used where different objects within a library have different protection requirements. Object security may be necessary where the library structure does not support security requirements or may be used to implement exceptions to the general library authorization structure.

4.7.3 Menu Security

Menu security is related to limiting a user′s capabilities and restricting him to a predefined secured environment. The user′s initial program and menu structure will restrict him to the functions and objects he is allowed to use. The user is not allowed to issue commands, therefore, all functions he requires should be available from a menu. Also, every menu he accesses should only have options that he needs.

4.7.4 Overall Recommendations

The overall recommendations given in this topic rely on one important principle: *simplicity*. If you keep your security design as simple as possible, you will be able to manage and audit your system security with ease.

We recommend that you use the different philosophies in combination. The following steps should be followed when designing the overall security scheme:

1. Move from general to specific:

    - Plan security for libraries first. Deal with individual objects only when necessary.

- Plan public authority first, follow by group authority and individual authority. Refer to 4.6, "*PUBLIC Authority" on page 4-6.

2. Library Security
   Libraries should be designed in a way that objects contained in a library have identical or at least similar protection requirements. Authorizations to libraries should then be established as a first step. We recommend that explicit authorizations be defined for all production libraries; it may be acceptable to protect test libraries through *PUBLIC authorization.

3. Object Security
   Specific object authorities should only be defined to handle exceptions. Exceptions exist where few objects within a library have more stringent protection requirements than defined for the library, and where temporary access must be granted. Otherwise, the public authority to the object should be adequate.

4. Menu Security
   We recommend use of the limited capability approach where appropriate to complement library and object security. We do not believe that menu security alone is a viable alternative. This recommendation is based on the fact that library and object security is enforced by the system, while initial programs, menus, and so on are largely user-designed and therefore more likely to have security exposures.

4.7.5 Recommended Protection Techniques

In applying object security, there are two issues: method of defining authorization, and object ownership.

Individual or Group Authorization

We recommend the use of multiple groups in the correct sequence as a general rule.

Individual or Group Ownership

In a production environment you get better control of security when a single user profile owns all the objects in an application.

In the test environment it may be an advantage to make programmers members of group profiles and give the group profile ownership of all objects.

When the objects are moved from the test environment to the production environment the ownership must be changed.

4.7.6 Authorization Lists

Where group profiles do not offer the required granularity, the use of authorization lists is recommended. They offer save and restore performance advantages over specific object authorization (based on the implementation - usually not visible to the end user) and functionally have the advantage that they survive the deletion of their related objects.

An authorization list could be established to secure all libraries within an application (and other major objects where appropriate) during the initial security implementation.

4.7.7 Logical Files

For access to critical files, logical files should be used. This way the owner of the file can authorize other users to specific fields (for example, address and phone number, but not salary) or specific records (for example, amounts less than $500) rather than the entire physical file. This is commonly known as field- or record-level security.

4.8 Authorization Search Sequence

There is a defined order for searching for authorization elements. It is very important to understand that the *first* authorization entry found (that matches the user and object) is taken. The exception to this is when using multiple groups. The authorities groups have are *not* accumulated when granting or revoking authorities. The authorities for all groups are accumulated. There may be other authorization matches for the user/object (which may be higher or lower authority than the first match) but they are not used. If a user has several authorizations to an object, the system does not take the highest authorization; it takes the first authorization, whatever level that may be.

4.9 Output Distribution

As in any computing environment, printed listings cannot be protected by the system after they are printed. It is very common to see a confidential report sitting in the printer output hopper waiting for the originator to pick it up. Manual distribution controls are required in this area, but the details will differ with every installation. In addition, security for spooled files *waiting* to be printed could be a possible exposure if not managed correctly.

4.9.1 Security in Output Queues

This section discusses security in output queues, and how to allow or disallow users to perform functions on output queues and on spooled files.

Possible Exposure

The AS/400 offers comprehensive facilities for spooled file management. The facilities are very easy to use, and as such could provide the opportunity for a user to access the confidential data of another user that is waiting to be printed. This exposure is greater if two companies share a common system, such as in a computer bureau environment.

Methods of Protection

There are several levels of security to an output queue. The definitions must be set in conjunction with the capabilities different users need to have:

- Working with all output queues
- Displaying the content of the spooled files on the output queue
- Working with the spooled files (change, delete and so on)

The level of authority to an output queue, and to the spooled files in the output queue, is determined by parameters in both the User Profile and in the output queue itself. Table 4-4 on page 4-10 summarizes the parameters which affect OUTQ security.

It must be emphasized that **a user with special authority *SPLCTL is able to perform all operations on all output queues, including their contents, regardless of other parameter settings in the output queue**. Therefore, only a very limited set of users should have *SPLCTL special authority if there is confidential data waiting to be printed.

Table 4-4 (Page 1 of 2). User Profile and Output Queue Parameters Affecting Security

Command	Parameter	Value	Meaning
CRTOUTQ Create Output Queue(1)	DSPDTA Display any file	*NO	A user cannot display, send, or copy spooled files owned by other users, unless the user has one of the following: • *JOBCTL special authority if the OPRCTL parameter is *YES, • *CHANGE authority to the output queue if the *AUTCHK parameter is *DTAAUT, • Ownership of the output queue if the *AUTCHK parameter is *OWNER.
		*YES	Any user having authority to read the queue can display, copy, or send the data of any spooled file on the queue.
		*OWNER	Only the owner of a spooled file can display, copy, send or move the file. If OPRCTL value is *YES, users with *JOBCTL special authority can hold, change, delete, and release output files but they cannot display, copy, send or move the spooled files. This is intended to allow operators to manage the entries on an output queue without being able to view the contents.
	OPRCTL Operator Controlled	*YES	A user with *JOBCTL special authority in the User Profile can control the queue and make changes to the spooled files on the queue, unless the DSPDTA value is *OWNER. If the DSPDTA value is *OWNER, *JOBCTL special authority does not allow the user to display, copy, send, or move spool files.
		*NO	This queue and its entries cannot be controlled or changed by users with *JOBCTL special authority unless they also have some other authority that overrides this(2)
	AUTCHK Authority to check. (3)	*OWNER	Specifies that only the owner of the output queue can control all the spooled files on the queue.
		*DTAAUT	Specifies that any user with read, add, and delete authority to the output queue can control all spooled files on the queue
	AUT Authority	*USE	*USE authority allows the user to perform basic operations on the output queue, such as send spooled files to the queue.
		*CHANGE	Allows the user to change the output queue description, and to control spooled files created by other users, if the queue was created with *DTAAUT specified for the Authority to check prompt (AUTCHK parameter).
		*ALL	Authority that allows the user to perform all operations on the output queue except those limited to the owner.
		*EXCLUDE	Authority that prevents the user from accessing the object, unless the user has some special authority.

Table 4-4 (Page 2 of 2). User Profile and Output Queue Parameters Affecting Security

Command	Parameter	Value	Meaning
CRTUSRPRF Create User Profile	SPCAUT Special Authority	*JOBCTL	A user with special authority *JOBCTL is able to change, display (DSPDTA(*OWNER) does not allow display), hold, release, cancel and clear all spooled files that are on an output queue (as well as jobs running or on the job queue). This is if the output queue is specified as Operator controlled (*YES).
		*SPLCTL	A user with special authority *SPLCTL is able to do everything with all output queues, regardless of other parameter settings in the OUTQ. Therefore, only the security officer should have *SPLCTL special authority.

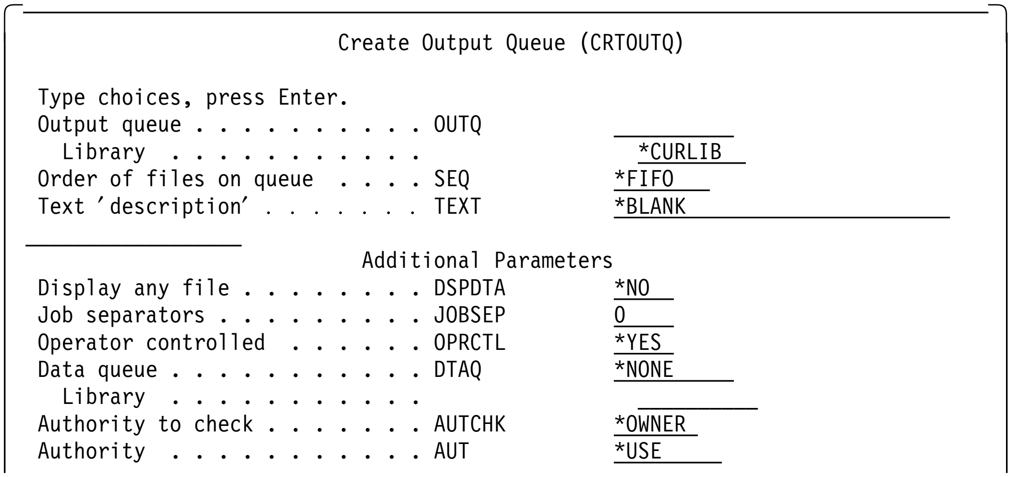
Note:

1. See Figure 4-2 for the CRTOUTQ command
2. See Table 4-5 on page 4-12.
3. The parameter AUTCHK (Authority to check) specifies what type of authorities are needed to the output queue that allows the user to control all the files on the queue. Users with some other authority may also be able to control the output files (see Table 4-5 on page 4-12).

Beware that creating an OUTQ in a library that has authority of PUBLIC *EXCLUDE does *not* prevent users with *JOBCTL or *SPLCTL from viewing or manipulating the spooled files. Similarly, a user with *ALLOBJ authority will not be prevented from manipulating an output queue if the object authority to the output queue is *EXCLUDE.

Table 4-5 on page 4-12 shows the possible combinations of User Profile and output queue parameters, with the resultant general capabilities for the user working with output queues. Refer to the *AS/400 Security Reference* for more information.

Due to the number of combinations, it is possible to create the desired environment, to meet the needs of the selected users. Figure 4-2 shows the create output queue (CRTOUTQ) command.



```
                     Create Output Queue (CRTOUTQ)

 Type choices, press Enter.
 Output queue . . . . . . . . . . OUTQ         __________
   Library  . . . . . . . . . . .                *CURLIB
 Order of files on queue  . . . . SEQ          *FIFO
 Text 'description' . . . . . . . TEXT         *BLANK

                          Additional Parameters
 Display any file . . . . . . . . DSPDTA       *NO
 Job separators . . . . . . . . . JOBSEP       0
 Operator controlled  . . . . . . OPRCTL       *YES
 Data queue . . . . . . . . . . . DTAQ         *NONE
   Library  . . . . . . . . . . .                __________
 Authority to check . . . . . . . AUTCHK       *OWNER
 Authority  . . . . . . . . . . . AUT          *USE
```

Figure 4-2. Create Output Queue Command

To view the parameter settings of an existing output queue use the ″work with output queue description″ command:

```
WRKOUTQD OUTQ(library/outq)
```

To see the object authority assigned to the output queue, use the ″display object authority″ command:

```
DSPOBJAUT OBJ(library/outq) OBJTYPE(*OUTQ)
```

User Profile Parameter	**OUTQ Parameter**				**Capabilities (see notes)**			
SPCAUT	*OPRCTL*	*Object Authority*	*DSPDTA*	*AUTCHK*	*1*	*2*	*3*	*4*
***JOBCTL**	*YES	N/A	N/A	N/A	Y	N(a)	Y	Y
***SPLCTL**	N/A	N/A	N/A	N/A	Y	Y	Y	Y
***JOBCTL** **OR** ***SERVICE** ***SECADM** ***SAVSYS**	*NO N/A	*CHANGE	N/A	*DTAAUT	Y	Y	Y	Y
			*YES	*OWNER	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)
		*USE	*YES	N/A	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)
		*EXCLUDE	N/A		N	N	N	N
***ALLOBJ**	N/A	N/A	N/A	*DTAAUT	Y	Y	Y	Y
			*YES	*OWNER	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)

Note:

1. User is able to look at the output queue (WRKOUTQ)
2. User is able to look at the content of spooled files in the output queue
3. User is able to add new spooled files to the output queue
4. User is able to change and delete spooled files

Y - indicates the function may be performed.
N - indicates the function may not be performed.
N/A - indicates that the function is not applicable,
N(a) - indicates that the function may not be performed,
unless the user is the owner of the object in the output queue.

Table 4-5. Output Queue Security. Combinations of User Profile Special Authority parameter and Output Queue parameters with resultant OUTQ capabilities.

4.9.2 How to Secure a Typical Printing Environment

This topic discusses how to secure a typical printing environment that can be found in many AS/400 installations. In this environment a printer is assigned to a specific user or to a group of users. They are limited users (most of them only using menus), and with no special authorities. Generally, the conditions that you can find are:

1. Users are to be prohibited from seeing (or copying) output from other users on a shared output queue.

2. Users are to be able to start printer writers by themselves (without operator intervention).

These two conditions conflict with each other, because if you want no one else to be able to see your own spool files, on a shared output queue, you should specify DSPDTA(*NO) and AUTCHK(*DTAAUT) in the output queue description in which your spool files reside.

On the other hand, if you want a user to be able to start a printer writer using a shared output queue, you should grant *READ, *ADD, and *DELETE authorities to the output queue for every user that shares it, giving them, in this way, the capacity to see someone else′s spool files.

Shared Output Queue

There is another way to secure the spool files in a shared output queue. You might specify DSPDTA(*OWNER) in the output queue description, but then only the owner of the output queue will be able to start a printer. To start the print writer you must have one of the following:

- *SPLCTL special authority in the user profile
- *JOBCTL special authority in the user profile and OPRCTL(*YES) in the output queue
- If the output queue has AUTCHK(*OWNER), you must be the owner of the output queue
- If the output queue has AUTCHK(*YES), you must have ADD/READ/DELETE authority to the output queue
- Ownership of the output queue

When an output queue has DSPDTA(*OWNER) and AUTCHK(*DTAAUT) users can be authorized to start writers to the output queue, but can only see the content of their own spooled files. They can, however, delete the spooled files of other users sharing the output queue.

Note: Output produced by a program that adopts its owner′s authority is owned by the user executing the program.

Non-Shared Output Queue

We suggest the following solution to this problem, which might apply in your environment.

You will have to create an output queue for every user and setup their user profiles to use their own output queues. In this way, all the spool files generated by a user will reside in his own output queue. Each one of the output queue should be created issuing:

```
CRTOUTQ (OUTQUSRx) DSPDTA(*OWNER) OPRCTL(*YES) AUTCHK(*DTAAUT)
                   TEXT(′Data queue for user x′)
```

*PUBLIC authority for the output queue should be *READ. This is the only authority needed for any user to add a spool file to an output queue. Every user will need *READ, *ADD and *DELETE authorities to their own output queues.

With this environment, users will be able to start any printer writer using their own output queues, for example if USR1 want to print his spool files using PRT3, he can issue the following command:

```
STRPRTWTR DEV(PRT3) OUTQ(OUTQUSR1)
```

Let′s look at another example:

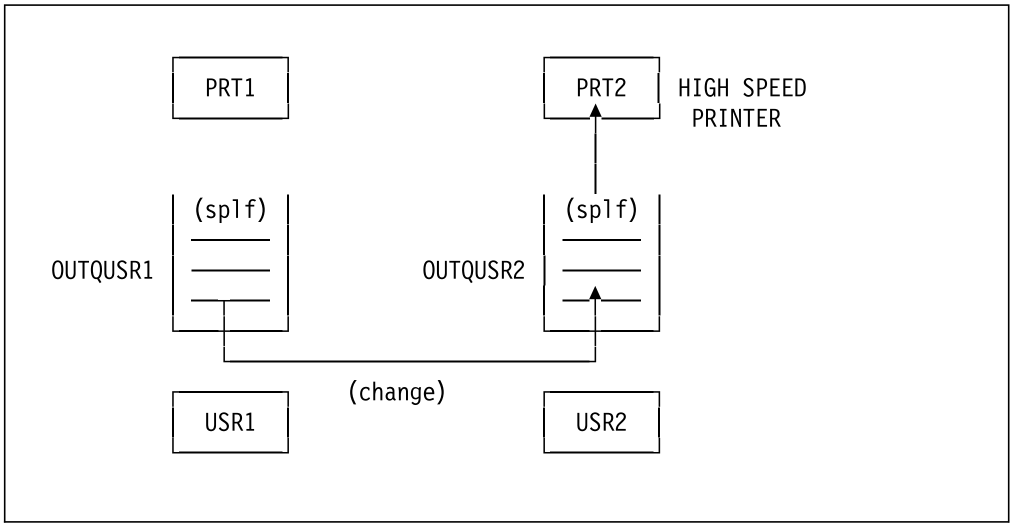


Figure 4-3. Example of a Printing Environment

In this case we have two users, each one has his own output queue. USR1 can use PRT1 to print his spool files, but he could have a big spool file that must be printed using a high speed printer (PRT2). He has two choices; the first is to start that printer writer using his own output queue. He can use this command:

```
   STRPRTWTR DEV(PRT2) OUTQ(OUTQUSR1)
```

But if USR2 is using his printer (PRT2), USR1 will get the next message: `CPF3310 writer PRT2 already started`.

The other way is to change his spool file to OUTQUSR2 output queue. Even if he changes his spool file to another user′s output queue, nobody, except him, is able to display, send or copy his output file. But note that USR2 will be able to hold it or delete it.

If you don′t want each user to have his own output queue, you can solve this problem creating a special CL program which adopts any user′s with *JOBCTL authority. This CL program should let you choose a printer and start its writer. The shared output queues should be created issuing:

```
   CRTOUTQ (SHRDOUTQ) DSPDTA(*OWNER) OPRCTL(*NO) AUTCHK(*DTAAUT)
                      TEXT(′Shared data queue′)
```

4.10 *SAVSYS Special Authority Considerations

A user with Save system (*SAVSYS) special authority, represents a potential security exposure to your system. *SAVSYS special authority gives the user the authority to save, restore, and free storage for *all* objects on your system, whether or not that user has object existence authority to the objects.

The user with *SAVSYS special authority is able to:

- Save an object and take it to another AS/400 to be restored
- Save an object and display (or dump) the tape to view the data
- Save and object and free storage, in this way deleting the data portion of an object (for applicable object types only)

- Save a document and delete it

As you see, giving *SAVSYS special authority to a user will enable him to do certain operations that could be exposures. That is why you must carefully evaluate the need of this special authority to a user and, if you grant it, keep track of the users who have it and periodically review their operations.

4.10.1 Other Save and Restore Considerations

Availability is a key consideration for any system. A fundamental requirement for availability are save and restore functions. Save and restore are important for everything from minor problems (for example, restore a source program where the updating got out of hand) to the complete loss of the computing facility. The restore may be to the same system or to another system, perhaps in a remote location.

The following table shows you some of the most common save and restore commands, and the authority required to the objects you want to save or restore with these commands.

	Authority Required											
Command	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
SAVCHGOBJ	x		x				x	x	x			
SAVLIB	x		a		a		x	x	x			
SAVOBJ	x		x				x	x	x			
SAVSYS												x
SAVSTG												x
RSTAUT												x
RSTCFG		x		a	a	x						
RSTLIB		x		a	a	x	x	x		x	x	
RSTOBJ		x		a	a	x	x	x		x	x	
RSTUSRPRF												x

Note: a = either or both of the indicated authorities

1. Object Existence authority for each object saved/restored
2. Object ownership or object existence
3. Read authority for the library
4. Read and Add authority for the library
5. Library ownership for the library
6. Add authority for the User Profile (for a new object)
7. Use authority for the device description (if tape or diskette)
8. Use authority for the savefile (if a savefile is used)
9. Add authority for the savefile (if a savefile is used)
10. Use authority for the CRTSAVF command (create savefile)
11. Use authority for saved-from library (if VOL(*SAVVOL))
12. SAVSYS authority required

Table 4-6. Authority for Key Save and Restore Commands

The user of save and restore commands must have *SAVSYS authority in his user profile or have the shown combinations of authorities. If he has *SAVSYS he may save any object without needing the other authorities shown here. For restrictions on the other save and restore commands (SAVS36F, SAVS36LIBM, SAVDLO, SAVLICPGM, RSTDLO, RSTLICPGM, RSTS36F, RSTS36FLR, and RSTS36LIBM) see appendix in the *AS/400 Security Reference*.

Save/Restore Device Control: You should consider controlling the authority to tape and diskette devices to ensure that only authorized users are able to perform save and restore operations. This same control should be extended to save files, if they are used in your site.

4.11 Securing Commands Related to Access Control

There are certain commands on the AS/400 that you should control access to because they represent an exposure to your system′s security, availability and normal operation.

Note that these commands are contained in multiple libraries (QSYS38, QSYSV2R1M0, QSYSV2R1M1, and all other libraries with the naming convention of QSYSxxxxxx). To be completely protected, all versions of these commands should be secured with *PUBLIC *EXCLUDE.

We recommend you to carefully decide which users will be able to execute the following commands:

System values and network attributes

- CHGNETA - Change Network Attributes
- CHGSYSVAL - Change System Value

Directory Entries

- ADDDIRE - Add Directory Entry

Changing or deleting line, controller or device descriptions

- CHGLINxxx - Change Line Descriptions
- DLTLINxxx - Delete Line Descriptions
- CHGCTLxxx - Change Controller Descriptions
- DLTCTLxxx - Delete Controller Descriptions
- CHGDEVxxx - Change Device Descriptions
- DLTDEVxxx - Delete Device Descriptions

Controlling communications status

- VRYCFG - Vary Configuration

Working with subsystem descriptions

- CHGSBSD - Change Subsystem Description
- DLTSBSD - Delete Subsystem Description

Working with job descriptions

- CHGJOBD - Change Job Description
- DLTJOBD - Delete Job Description

Saving and restoring information

- RSTxxx - Restore xxx
- SAVxxx - Save xxx

Managing libraries

- DLTLIB - Delete Library
- CLRLIB - Clear Library
- CRTLIB - Create Library
- CHGLIB - Change Library

Managing Files

- DLTF - Delete File
- CLRPFM - Clear Physical File Member

Using Data File Utility/400 (DFU)

- STRDFU - Start Data File Utility
- UPDDTA - Update Data

Using Query/400

- STRQRY - Start Query

Using SQL/400

- STRSQL - Start Structured Query Language

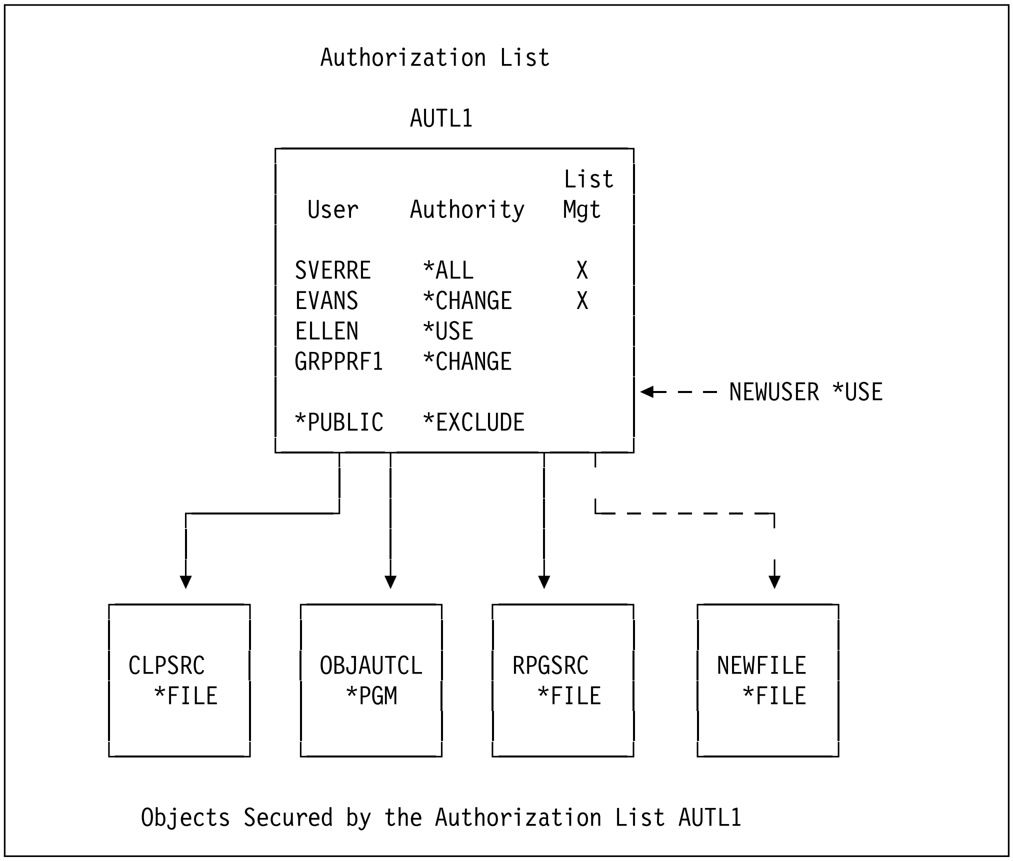
Using powerful system functions

- STRCMNTRC - Start a Communication Trace for a specified line
- STRSST - Start System Service Tools
- PWRDWNSYS - Power Down System

Chapter 5. Authorization Lists

AS/400 security is a combination of the best features of the System/36 and System/38. Authorization lists are one of the features that OS/400 inherited from the S/36 that allows the user to simplify the security management and reduce the system backup time. This function is new to the S/38 users. The S/36 users will see that the use of authorization lists has been expanded from use only in S/36 office to most objects on the AS/400 System.

A frequent error in security planning is securing the objects that do not need to be protected. If an object does not need to be protected, the most efficient from a system performance and security management standpoint is use of *PUBLIC authority. Authorization lists and/or private authorizations should be used when an object needs to be secured.



User	Authority	List Mgt
SVERRE	*ALL	X
EVANS	*CHANGE	X
ELLEN	*USE	
GRPPRF1	*CHANGE	
*PUBLIC	*EXCLUDE	

Figure 5-1. Authorization List and Objects

An authorization list references both user profiles and the resources (objects). These user profiles are authorized to the objects on the authorization list. The authorization list AUTL1, shown graphically in Figure 5-1, has four user profiles and a *PUBLIC authority of *EXCLUDE. The user profiles are authorized to the three objects secured by the list. The file NEWFILE and the user NEWUSER will be added to the authorization list. We will discuss how to create the authorization list and add users and objects later.

All user profiles on the authorization list are authorized to an object in one operation. The list of user profiles are authorized to the file NEWFILE by simply specifying the name of the authorization list (AUTL1) when the file is created. This single operation requires less effort than authorizing the individual user profiles. The use of authorization lists rather than individual user authorities will also improve the system backup time. A similar one-step operation can remove an authorization list from an object. This step, in effect, removes the authority to the object from all the user profiles on the authorization list.

Adding a user profile to an authorization list will authorize the user profile to all the objects secured by the authorization list. Adding the user profile NEWUSER to the authorization list AUTL1 gives this user profile authority to the objects CLPSRC, OBJAUTCL, RPGSRC and (the new object) NEWFILE.

The user profiles on an authorization list can be individual user profiles or group profiles. In Figure 5-1 on page 5-1, the profile GRPPRF1 is a group profile that has multiple members. Since the group profile is on the authorization list, each member of the group is authorized with *CHANGE authority. If profiles ELLEN, NEIL, TRACEY and TROY were the members of GRPPRF1, they have *CHANGE authority to the objects. When a user profile that is a member of the group is also on the authorization list, the individual user profile authority is used instead of the group profile. Because the user profile ELLEN is authorized on the list AUTL1, the authority for user profile ELLEN is *USE.

To allow easy identification of a group profile, your organization should establish a naming convention such as GRPxxxx or DPTxxxx to identify the group profiles. Using a naming convention allows the instant recognition of a group profile when profiles are shown or listed. A group profile usually indicates that multiple users are authorized to the object. For more information about group profiles, see Chapter 3, “User Profiles and Group Profiles” on page 3-1.

5.1 Creating an Authorization List

Authorization lists are created by the CRTAUTL command. The authorization list AUTL1 is created by the following command:

```
CRTAUTL AUTL(AUTL1) AUT(*EXCLUDE) TEXT(′Sample Authorization List′)
```

The create command will place the owner, SVERRE in this example, on the authorization list with *ALL and *AUTLMGT authority.

The AUT parameter of the CRTAUTL command defines the public authority on the authorization list. This public authorization list is used when the public authority on the object is specified *AUTL and there is no authority for the user profile or the group profile for the user. When an object has public authority, the public authority on the authorization list is not used.

Authorization lists are assigned a name that must be unique for the system. A good practice is to establish a naming convention for authorization lists where the first few characters indicate the area that owns the authorization list, followed by additional characters to make the name unique. Using this convention, the ″sales″ area would name the authorization list SLSAUTL1, the characters SLS indicating the authorization list is owned and managed by the ″sales″ area.

5.2 Adding Users to an Authorization List

The following commands add the users to the authorization list AUTL1:

```
ADDAUTLE AUTL(AUTL1) USER(ELLEN  )  AUT(*USE)
ADDAUTLE AUTL(AUTL1) USER(GRPPRF1)  AUT(*CHANGE)
ADDAUTLE AUTL(AUTL1) USER(EVANS  )  AUT(*CHANGE *AUTLMGT)
```

The capability to add or remove a user profile to or from an authorization list requires a special authority because adding a user profile is in effect authorizing that profile to every object secured by the authorization list.

To manage the user profiles on an authorization list requires one of the following authorities:

- The owner of the authorization list can add or remove user profiles and has full control over the authority of user profiles on the authorization list.

  Because of the additional control the owning user profile has, you should specifically consider the ownership of the authorization lists that secure sensitive objects. The security administrator or security officer profile should own the authorization lists that secure sensitive objects.

- The users with *ALLOBJ special authority also have full control over the users on an authorization list.

- The users with *AUTLMGT (authorization list management) authority can add or remove users on the authorization list, but can only grant users a subset of their authorities.

  For example, the user EVANS on the authorization list AUTL1 has *AUTLMGT and *CHANGE authority. The *AUTLMGT authority allows EVANS to add or remove users to or from the list and grant them *CHANGE or less authority. Only the authorization list owner or an *ALLOBJ user can grant *AUTLMGT authority. However, the users with *AUTLMGT authority can remove users who have *AUTLMGT authority and equal or subset of their authorities. Using this authority, user EVANS can add the user NEWUSER with the following command.

  ```
      ADDAUTLE AUTL(AUTL1) USER(NEWUSER)  AUT(*USE)
  ```

5.3 Assigning Objects to an Authorization List

An object can be assigned to an authorization list when the object is created or for an existing object, an authorization list can be assigned by using the grant or edit commands. These two methods are illustrated below:

- The create commands (CRTxxx) for some object types (*CMD, *DOC, *FILE, *FLR, *LIB and *PGM) allow an authorization list name to be specified in the AUT parameter. For example, in creating a CL program OBJAUTCL in library SVERRE, assign the program to the authorization list AUTL1 with the following command:

  ```
  CRTCLPGM PGM(SVERRE/OBJAUTCL) AUT(AUTL1)
  ```

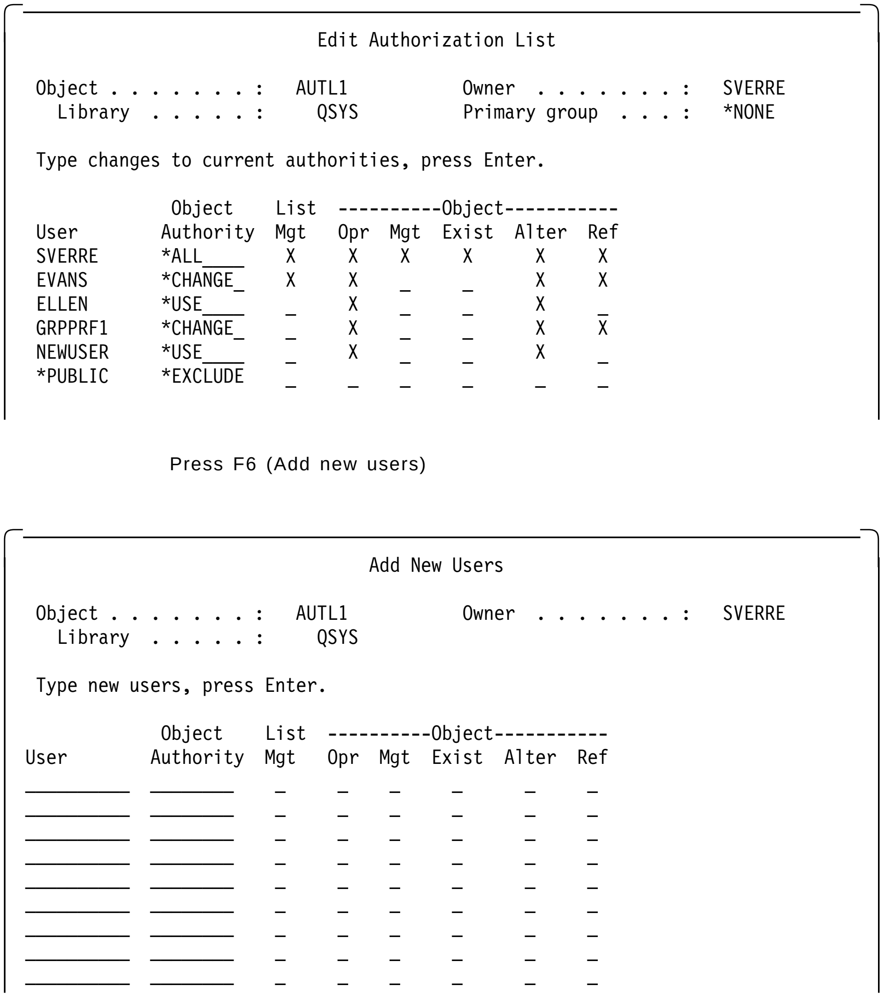
  The default when creating a document is to secure the document using the authorization list of the folder where the document is stored.

- If the object already exists, it can be assigned to an authorization list using the Grant Object Authority (GRTOBJAUT) command. Only the owner of the object, a user with all object (*ALLOBJ) special authority, or a user with all (*ALL) authority to the object, can add an object to the authorization list AULT1 as follows:

```
GRTOBJAUT OBJ(SVERRE/CLPSRC) OBJTYPE(*FILE) AUTL(AUTL1)
GRTOBJAUT OBJ(SVERRE/RPGSRC) OBJTYPE(*FILE) AUTL(AUTL1)
```

The two object types that cannot be protected by an authorization list are *User Profile* and *Authorization List*.

The edit (EDTOBJAUT or EDTDLOAUT) commands provide an easy-to-use interactive interface that perform the equivalent function as the commands described previously. The EDTAUTL command provides an interactive interface to manage users on an authorization list as shown in Figure 5-2. The authority of users can be changed by modifying the Object Authority column or adding or removing an X in the Object or Data authority columns (you will need to press F11 to show these columns for changing). Blanking the authority for a user will remove the user from the list. If F6 (Add new users) is pressed, the ″Add New Users″ screen is displayed and additional users can be added to an authorization list.



```
                         Edit Authorization List

 Object . . . . . . . :   AUTL1          Owner  . . . . . . . :   SVERRE
   Library  . . . . . :     QSYS         Primary group  . . . :   *NONE

 Type changes to current authorities, press Enter.

             Object    List  ----------Object-----------
 User        Authority Mgt   Opr  Mgt  Exist  Alter  Ref
 SVERRE      *ALL____   X     X    X     X      X     X
 EVANS       *CHANGE_   X     X    _     _      X     X
 ELLEN       *USE____   _     X    _     _      X     _
 GRPPRF1     *CHANGE_   _     X    _     _      X     X
 NEWUSER     *USE____   _     X    _     _      X     _
 *PUBLIC     *EXCLUDE   _     _    _     _      _     _
```

Press F6 (Add new users)

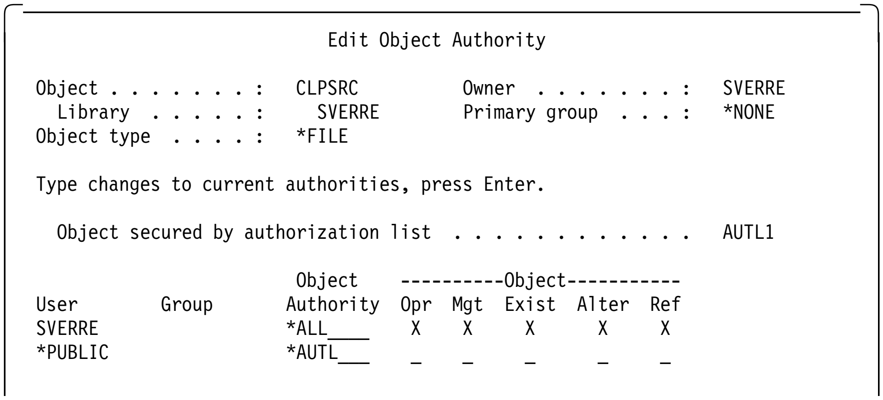
```
                              Add New Users

 Object . . . . . . . :   AUTL1          Owner  . . . . . . . :   SVERRE
   Library  . . . . . :     QSYS

 Type new users, press Enter.

            Object    List  ----------Object-----------
User        Authority Mgt   Opr  Mgt  Exist  Alter  Ref
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
__________  ________   _     _    _     _      _     _
```

Figure 5-2. EDTAUTL AUTL(AUTL1). Changing User Authority for Authorization List AUTL1

The EDTOBJAUT command provides an interactive interface to manage authority for an object. Figure 5-3 on page 5-5 is the screen displayed for the following command:

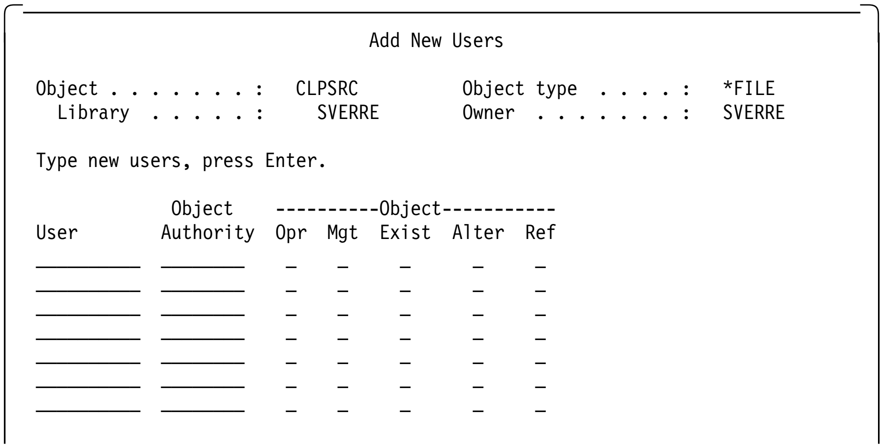
```
EDTOBJAUT OBJ(CLPSRC) OBJTYPE(*FILE)
```

This interactive screen has the same operational characteristics as the interactive interface for authorization lists. Changing the authorization list for an object can be done by entering the change in the field that shows the name of the authorization list.



```
                            Edit Object Authority

 Object . . . . . . . :   CLPSRC         Owner  . . . . . . . :   SVERRE
   Library  . . . . . :     SVERRE       Primary group  . . . :   *NONE
 Object type  . . . . :   *FILE

 Type changes to current authorities, press Enter.

   Object secured by authorization list  . . . . . . . . . . . .  AUTL1

                          Object    ----------Object-----------
 User        Group       Authority  Opr  Mgt  Exist  Alter  Ref
 SVERRE                  *ALL____    X    X     X      X     X
 *PUBLIC                 *AUTL___    _    _     _      _     _
```

Press F6 (Add new users)



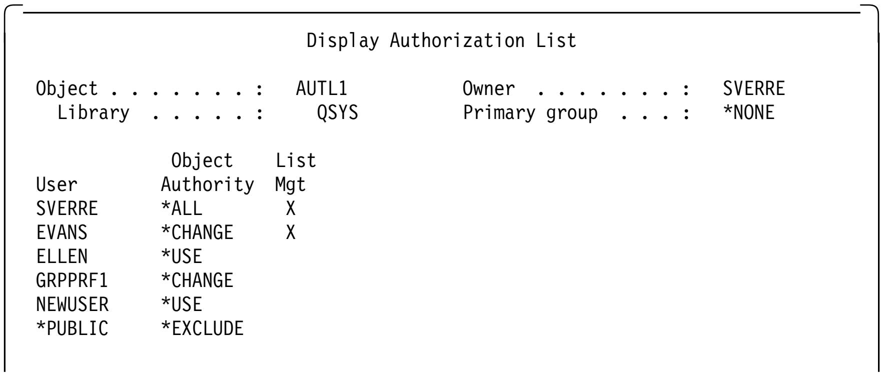
```
                                Add New Users

 Object . . . . . . . :   CLPSRC         Object type  . . . . :   *FILE
   Library  . . . . . :     SVERRE       Owner  . . . . . . . :   SVERRE

 Type new users, press Enter.

             Object    ----------Object-----------
 User       Authority  Opr  Mgt  Exist  Alter  Ref
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
 __________ ________    _    _     _      _     _
```

*Figure 5-3. EDTOBJAUT OBJ(CLPSRC) OBJTYPE(*FILE). Changing the Authority for the File CLPSRC*

5.4 Display Users and Objects on an Authorization List

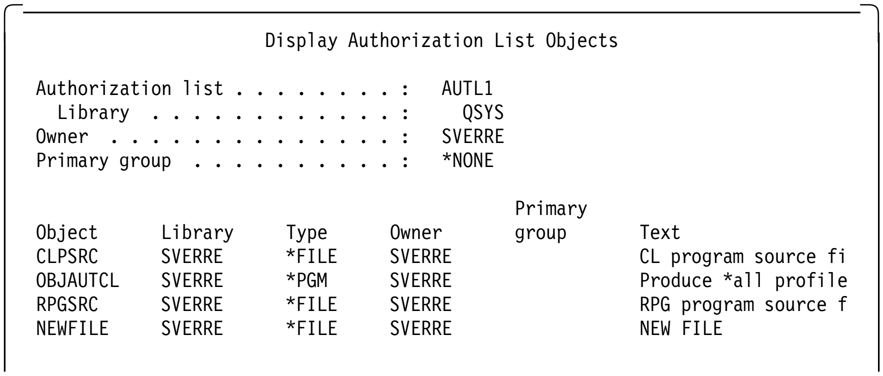
The DSPAUTL and DSPAUTLOBJ commands display the list of users or list of objects on an authorization list. Screens for these commands for authorization list AUTL1 are shown in figure Figure 5-4 on page 5-6. Press function key F15 (Display authorization list objects) from the screen of ″Display Authorization List″ or enter the DSPAUTLOBJ command to show the list of objects secured by the authorization list.

The screens for the edit and display commands look similar which makes it easy to learn the functions. The difference between display (DSP) and edit (EDT) commands is that display commands allow viewing of object attributes while the edit and work with commands allow viewing and modification of object attributes. This same strategy is used for the system display, edit, and work with commands for all objects.



```
                         Display Authorization List

Object . . . . . . . :   AUTL1          Owner  . . . . . . . :   SVERRE
  Library  . . . . . :     QSYS         Primary group  . . . :   *NONE

           Object    List
User       Authority  Mgt
SVERRE     *ALL        X
EVANS      *CHANGE     X
ELLEN      *USE
GRPPRF1    *CHANGE
NEWUSER    *USE
*PUBLIC    *EXCLUDE
```

Press F15 (Display authorization list objects)



```
                     Display Authorization List Objects

Authorization list . . . . . . . . :   AUTL1
  Library  . . . . . . . . . . . . :     QSYS
Owner  . . . . . . . . . . . . . . :   SVERRE
Primary group  . . . . . . . . . . :   *NONE

                                                  Primary
Object      Library     Type      Owner       group       Text
CLPSRC      SVERRE      *FILE     SVERRE                  CL program source fi
OBJAUTCL    SVERRE      *PGM      SVERRE                  Produce *all profile
RPGSRC      SVERRE      *FILE     SVERRE                  RPG program source f
NEWFILE     SVERRE      *FILE     SVERRE                  NEW FILE
```

Figure 5-4. DSPAUTL AUTL(AUTL1) and DSPAUTLOBJ AUTL(AUTL1). Users and Objects on an Authorization List

The DSPAUTL and DSPAUTLOBJ commands support an outfile so that either the users or objects secured by an authorization list can be retrieved into a database file. This outfile function can be used to back up a single authorization list. (An example of this in use is shown later in 5.12, “Managing Authorization Lists Between Systems” on page 5-12.)

5.5 Authorization List Implementation Detail

The representation of an authorization list as a list of users and objects is a conceptual representation. An authorization list does *not* contain a list of users ″on the authorization list″. There are no users ″on the authorization list″, rather the users are authorized to the authorization list object. The user profile for users that are ″on the authorization list″ contains an authorization entry for the authorization list. Like all objects, the header of an authorization list will contain the *PUBLIC and owner′s authority. The conceptual representation and actual implementation are shown in the Figure 5-5 on page 5-7. On the left is the conceptual representation. The actual implementation is shown on the right. The *PUBLIC and owner′s authority (EVANS in this example) are stored in the object header of the authorization list.

When a user profile is authorized to an object, the user profile contains the virtual address of the object and the user's authority to the object. The system checks for a user's authority to an object by searching for the virtual address of the object in the user profile. When an object has an authorization list, the system also searches for the address of the authorization list in the user profile.

An authorization list does contain a list of objects. However, this list of objects is not used for access control decisions. The list of object is used to display the list of objects that are secured by the authorization list. When an object is secured by an authorization list, the virtual address of the authorization list is stored in the object header of the object.

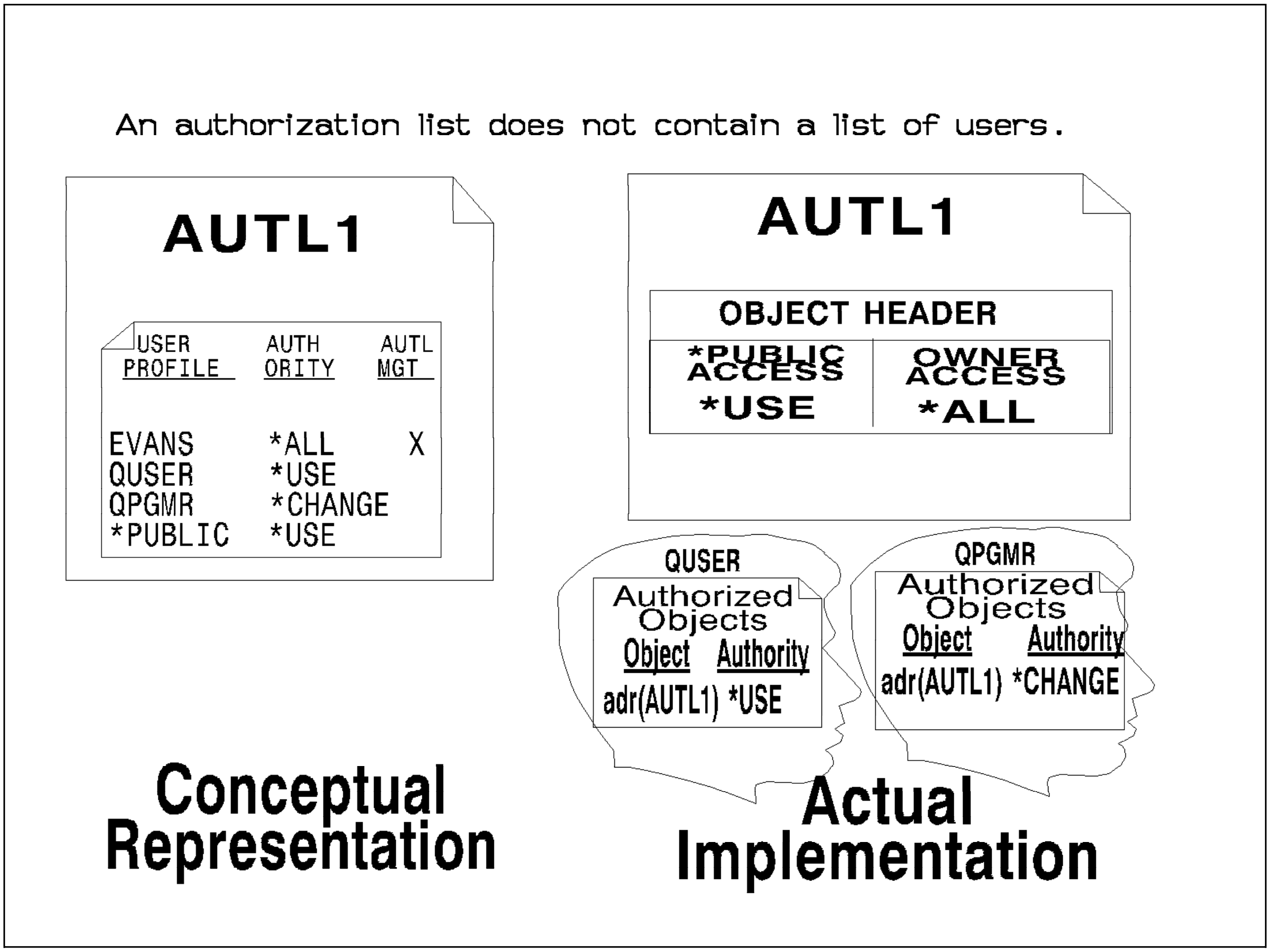


Figure 5-5. Implementation of Authorization Lists

5.6 Save/Restore Considerations

Authorization lists reside in library QSYS and are saved by a SAVSYS (Save System) or SAVSECDTA (Save Security Data). Authorization lists are restored by the RSTUSRPRF (Restore User Profiles) command when all user profiles are restored. The users and their authorities are restored to the authorization list by the RSTAUT (Restore Authority) command. There are no interfaces to save/restore an individual authorization list.

Saving an object that is secured by an authorization list will save the name of the authorization list with the object. If the object is restored to the *same* system where the save occurred, the object will be reattached to the authorization list during the restore operation. This automatic association of object to the authorization list does not happen on the restore to a *different* system because the named authorization list may be used for a different purpose on the other system and this prevents any potential security breach. Installations with multiple AS/400 systems have requested an option to reattach the object to the authorization list when restored on a system different from the one where the save occurred. This would be useful when organizations want to manage the security on the different systems in a similar manner. The programs illustrated in 5.12, "Managing Authorization Lists Between Systems" on page 5-12 provides a method to transfer objects between systems and maintain authorization lists.

5.7 Recovering from a Damaged Authorization List

When an object is secured by an authorization list and the authorization list becomes damaged, access to the object is limited to users that have all object (*ALLOBJ) special authority.

To recover from a damaged authorization list, two steps are required:

1. Recover users and their authorities on the authorization list.
2. Recover the association of the authorization list with the objects.

These steps must be done by a user with *ALLOBJ special authority.

5.7.1 Recovering the Authorization List

If users′ authorities to the authorization list are known, simply delete the authorization list, create the authorization list again, and then add users to it.

If it is not possible to create the authorization list again because you do not know all the user authorities, the authorization list can be restored and the users restored to the authorization list using your last SAVSYS or SAVSECDTA tapes. To restore the authorization list, do the following:

1. Delete the damaged authorization list using the Delete Authorization List (DLTAUTL) command.
2. Restore the authorization list by restoring user profiles:

   ```
   RSTUSRPRF USRPRF(*ALL)
   ```

3. Restore users′ private authorities to the list using the RSTAUT command.

5.7.2 Recovering the Association of Objects to the Authorization List

When the damaged authorization list is deleted, the objects secured by the authorization list need to be added to the new authorization list. Do the following:

1. Delete the QRCL library and the QRCLAUTL authorization list. (They are recreated by the RCLSTG command). Get rid of old stuff.
2. Find the objects that were associated with the damaged authorization list using the Reclaim Storage (RCLSTG) command. Reclaim storage assigns the objects that were associated with the authorization list to the QRCLAUTL authorization list.

3. Use the Display Authorization List Objects (DSPAUTLOBJ) command to list the objects associated with the QRCLAUTL authorization list.

4. Use the Grant Object Authority (GRTOBJAUT) command to secure each object with the correct authorization list:

```
GRTOBJAUT OBJ(library name/object name) AUTL(authorization list name)
```

Note: If a large number of objects are associated with the QRCLAUTL authorization list, create a database file by specifying OUTPUT(*OUTFILE) on the DSPAUTLOBJ command. You can write a CL program to run the GRTOBJAUT command for each object in the file. See 5.12, "Managing Authorization Lists Between Systems" on page 5-12 for more information.

5.8 Authority Search

When a user has (private) specific authority to an object but is also on the authorization list, the specific authority is used rather than the authorization list authority. The specific authority should be used for exceptions when a user has different authority than the authorization list. If the user profile EVANS should not have *CHANGE authority to the file NEWFILE as shown in Figure 5-1 on page 5-1, the following command can be used to grant *USE authority to the user profile EVANS.

```
GRTOBJAUT OBJ(SVERRE/NEWFILE) OBJTYPE(*FILE) USER(EVANS) AUT(*USE)
```

The specific authority overrides the authorization list. This allows additional flexibility. When an authorization list has the correct authority but there are some exceptions, specific authority can be used to handle the exceptions rather than create a new authorization list. If there are a number of objects or user profiles that need specific authority, a second authorization list should be created.

5.9 Performance Advantages of Authorization Lists

Authorization lists can decrease the number of authority entries and improve time required to back up system authorities. When a user profile is specifically authorized to an object, the system records this information in the user profile. These authority entries are saved by a SAVSYS or SAVSECDTA command. The time to perform the system backup increases with an increased number of authority entries.

When an object is secured by an authorization list, the system does not require an authorization entry for each object. Objects are associated with the authorization list by a system pointer from the object back to the authorization list. When an authorization list secures multiple objects, the number of authority entries in the system is reduced. This is especially true for multiple member database files. Figure 5-6 on page 5-10 compares the number of authority entries when specific authority or an authorization list is used.

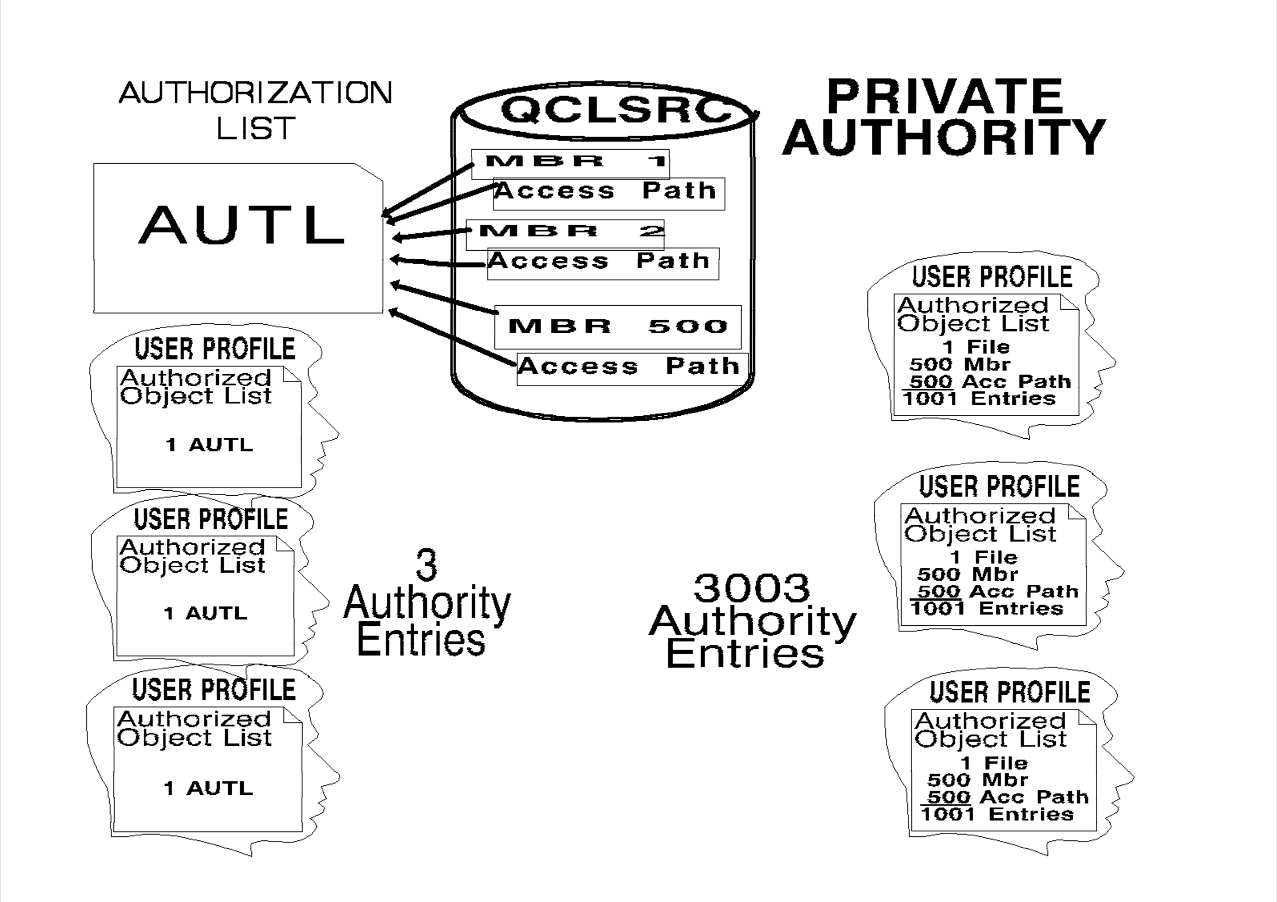


Figure 5-6. Comparison in Number of Authorizations

When a user profile is specifically authorized to a database file, the system replicates this authority to each of the members. When specific authorities are used with database files that have a large number of members, this replication of specific authority for each member can result in a large number of authorities in the system. The right half of Figure 5-6 shows the file QCLSRC with 500 members specifically authorized to three profiles. Each of the profiles is authorized to the file description object and each of the 500 members. This would result in 3*(1001) or 3003 authority entries.

The left half of Figure 5-6 illustrates the use of an authorization list rather than specific authorities. Rather than individual authority entries, the authority is associated using a pointer to the authorization list from the file description object and each of the 500 members. There are three authority entries for the authorization list AUTL. The reduction from 3003 to three authority entries reduces the time required to perform a system backup (SAVSYS). This use of authorization lists has significantly reduced the number of authorities and is even more significant when the same authorization list is used to secure other database files.

See Chapter 7, “Security Design For Performance” on page 7-1 for more information.

5.10 Comparing Authorization Lists to Group Profiles

Authorization lists and group profiles are both designed to simplify security management by grouping users and objects. Table 5-1 illustrates the differences and similarities between authorization lists and group profiles.

Table 5-1. Comparison of Authorization Lists and Group Profiles

AUTHORIZATION LISTS	**GROUP PROFILES**
Secure multiple objects	Secure multiple objects
Reduces authority entries by having one entry used for multiple objects or members of a database file. Pointers are used to associate authority, so a multiple member database file does not increase number of authority entries in the system.	Reduces authority entries since each group member does not need to be authorized. Specific authorization of group to a multiple member database file causes entries for each member.
Users on an authorization list can have a *different* authority	Members of a group share the *same* authority from the group
Object can be associated with a single authorization list	Multiple group profiles can be authorized to an object
A user can be on multiple authorization lists	A user can be a member of up to 16 group profiles
Authorization of a multiple member database file does not replicate authorities to each member	Authorization of group profile to a database file replicates authority to each member
No Equivalent function	Ownership of objects created by group members can be transferred to the group profile
Restore of object on same system will automatically attach to an authorization list	No equivalent function
Authorities specified by an authorization list are the same for all objects	Authorities specified by a group profile can be different for all objects
Private authority overrides other authority specified by an authorization list	Private authority overrides other authority specified by a group profile
Authority can be specified to be secured by an authorization list when the object is created	Authority can be specified when the object is created **Note:** Members of the group profile can be given authority at the time an object is created by the GRPAUT parameter in the profile of the user creating an object.
Can secure all object types **except User Profiles and Authorization Lists**	Can secure all object types
Association with object is deleted when the object is deleted	Association with object is deleted when the object is deleted
Association with object is saved when the object is saved	Association with object is not saved when the object is saved, but is saved when the user profile is saved

Another performance benefit of using authorization lists to secure multiple member database files is the time reduction to add or change user authorization. When specific authorities are used, the system must add or change entries for each member resulting in more processing time. When an authorization list is used, there are no member level operations required because the pointer from the object to the member is already existed.

A frequently asked question is, which is better, authorization lists or group profiles? The recommendation is to use both features. When you enroll users, assign them a group profile, but have objects they create owned by the user profile. The group profile name can be used on authorization lists and the group profile authority is available to group members.

Authorization lists offer the advantage of allowing different authority for different users. Pointers associate objects and authorization lists reducing the number of authorization records and the system back up time. The reduction in authorization records is more significant using authorization lists than by group profiles. Another advantage that authorization lists offer is the automatic association when objects are restored on the same system.

5.11 Limitations of Authorization Lists

Authorization lists can be used for objects stored in library QSYS, but special considerations are needed for a total system recovery for these objects. In the event the system must be reloaded, the objects in library QSYS are not attached to an authorization list. The objects are restored with the install, but the authorization lists are not restored until later when you perform a RSTUSRPRF (Restore User Profiles). Because the objects in library QSYS are restored before the authorization lists, the objects are not associated with the authorization lists.

Special planning is required as part of the system backup procedure to reattach objects in library QSYS. Prior to the system backup, a database file that lists all the objects on authorization lists must be produced. This can be done using the program ALLAUTL1 described in 5.12, “Managing Authorization Lists Between Systems.”

If you need to perform a total system restore, the objects in library QSYS can be reattached to their authorization list after the authorization lists have been restored (RSTUSRPRF). The information in the file can be used to attach objects back to the authorization lists, The program FIXAUTL1 in 5.12, “Managing Authorization Lists Between Systems” illustrates use of this file to reattach objects to authorization lists.

5.12 Managing Authorization Lists Between Systems

The following programs can be used to attach authorizations lists when objects are restored on a different system or when objects are restored in library QSYS because of a total system rebuild. The security officer runs the command ALLAUTL before the save operation which will create a database file for all objects on all authorization lists. The database file is then saved and restored to the target system. The command FIXAUTL will attach objects to authorization lists.

5.12.1 ALLAUTL1 - List All Objects on AUTL

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
  100             PGM       PARM(&PARM1)
  200  /*******************************************************************/
  300  /* ALLAUTL1-- This program creates a database file with all of the    */
  400  /*            authorization list object names.   This database file  */
  500  /*            can be as input to  the FIXAUTL1 program to associate   */
  600  /*            objects on an authorization list when restored on a     */
  700  /*            system that was not the same system used to save the    */
  800  /*            objects                                                 */
  900  /* INPUT  --  Name of the OUTFILE                                     */
  700  /* OUTPUT --  Name of the file containing list of objects and their   */
  800  /*            associated authorization list                           */
 1100  /* NOTE   --  Program MUST be run by a user with *ALLOBJ authority    */
 1200  /*******************************************************************/
 1300             DCL       VAR(&MSGID) TYPE(*CHAR) LEN(7)
 1400             DCL       VAR(&MSGDTA) TYPE(*CHAR) LEN(50)
 1500             DCL       VAR(&MSGF) TYPE(*CHAR) LEN(10)
 1600             DCL       VAR(&MSGLIB) TYPE(*CHAR) LEN(10)
 1700             DCL       VAR(&RTNTYPE) TYPE(*CHAR) LEN(2)
 1800             DCL       VAR(&ERROR) TYPE(*LGL)
 1900             DCL       VAR(&PARM1) TYPE(*CHAR) LEN(20)
 2000             DCL       VAR(&OUTFILE) TYPE(*CHAR) LEN(10)
 2100             DCL       VAR(&OUTLIB) TYPE(*CHAR) LEN(10)
 2200             DCL       VAR(&MBROPT) TYPE(*CHAR) LEN(10) +
 2300                         VALUE(*REPLACE)
 2400             DCLF      FILE(QADSPOBJ)
 2500         /********  START OF PROGRAM  ***************/
 2600             MONMSG    MSGID(CPF0000) EXEC(GOTO CMDLBL(ERROR))
 2700             CHGVAR    VAR(&OUTFILE) VALUE(%SST(&PARM1 1 10))
 2800             CHGVAR    VAR(&OUTLIB) VALUE(%SST(&PARM1 11 10))
 2900             DSPOBJD   OBJ(QSYS/*ALL) OBJTYPE(*AUTL) +
 3000                         OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTL)
 3100             OVRDBF    FILE(QADSPOBJ) TOFILE(QTEMP/AUTL)
 3200  READ:      RCVF
 3300             MONMSG    MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
 3400             DSPAUTLOBJ AUTL(&ODOBNM) OUTPUT(*OUTFILE) +
 3500                         OUTFILE(&OUTLIB/&OUTFILE) OUTMBR(*FIRST +
 3600                         &MBROPT)
 3700             MONMSG    MSGID(CPF6250 CPF9800) EXEC(GOTO CMDLBL(READ))
 3800             CHGVAR    VAR(&MBROPT) VALUE(’*ADD’)
 3900             GOTO      CMDLBL(READ)
 4000  EOF:       GOTO      CMDLBL(EXIT)
```

Figure 5-7 (Part 1 of 2). List All Objects in Authorization Lists

```
ERROR: /*************** ERROR HANDLING ROUTINE ***************/
            IF         COND(&ERROR) THEN(GOTO CMDLBL(EXIT))
            CHGVAR     VAR(&ERROR) VALUE('1')
RECEIVE:    RCVMSG     MSGTYPE(*ANY) MSGDTA(&MSGDTA) MSGID(&MSGID) +
                         RTNTYPE(&RTNTYPE) MSGF(&MSGF) +
                         MSGFLIB(&MSGLIB)
            IF         COND(&RTNTYPE *NE '15') THEN(DO) /* NOT +
                         EXCAPE MESSAGE    */
            SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                         MSGTYPE(*DIAG)
            GOTO       CMDLBL(RECEIVE)
            ENDDO
            SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                         MSGTYPE(*ESCAPE)
EXIT:       ENDPGM
                         * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 5-7 (Part 2 of 2). List All Objects in Authorization Lists

5.12.2 FIXAUTL1 - Add Objects to AUTL

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
            PGM        PARM(&PARM1)
 /********************************************************************/
 /* FIXAUTL1-- This program reads the database file with all of the  */
 /*            object names on authorization lists.    This program  */
 /*            will grant the objects to the specified authorization */
 /*            list.                                                 */
 /* INPUT  --  Name of the file containing list of objects and their */
 /*            associated authorization list                         */
 /* NOTE   --  Program MUST be run by a user with *ALLOBJ authority  */
 /********************************************************************/
            DCL        VAR(&MSGID) TYPE(*CHAR) LEN(7)
            DCL        VAR(&MSGDTA) TYPE(*CHAR) LEN(50)
            DCL        VAR(&MSGF) TYPE(*CHAR) LEN(10)
            DCL        VAR(&MSGLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&RTNTYPE) TYPE(*CHAR) LEN(2)
            DCL        VAR(&ERROR) TYPE(*LGL)
            DCL        VAR(&PARM1) TYPE(*CHAR) LEN(20)
            DCL        VAR(&OUTFILE) TYPE(*CHAR) LEN(10)
            DCL        VAR(&OUTLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&TOTAL) TYPE(*DEC) LEN(5 0) VALUE(0)
            DCL        VAR(&FAIL) TYPE(*DEC) LEN(5 0) VALUE(0)
            DCL        VAR(&TOTALC) TYPE(*CHAR) LEN(5)
            DCL        VAR(&FAILC) TYPE(*CHAR) LEN(5)
            DCLF       FILE(QADALO)
```

Figure 5-8 (Part 1 of 2). Add Objects to Authorization Lists

```
         /********  START OF PROGRAM  ***************/
            MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(ERROR))
            CHGVAR     VAR(&OUTFILE) VALUE(%SST(&PARM1 1 10))
            CHGVAR     VAR(&OUTLIB) VALUE(%SST(&PARM1 11 10))
            OVRDBF     FILE(QADALO) TOFILE(&OUTLIB/&OUTFILE)
READ:       RCVF
            MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
            CHGVAR     VAR(&TOTAL) VALUE(&TOTAL + 1.0)
            GRTOBJAUT  OBJ(&AOLIB/&AONAME) OBJTYPE(&AOTYPE) +
                         AUTL(&AOANAM)
            MONMSG     MSGID(CPF0000) EXEC(DO)
            CHGVAR     VAR(&FAIL) VALUE(&FAIL + 1.0)
            GOTO       CMDLBL(READ)
            ENDDO
            GRTOBJAUT  OBJ(&AOLIB/&AONAME) OBJTYPE(&AOTYPE) +
                         USER(*PUBLIC) AUT(*AUTL)
            GOTO       CMDLBL(READ)
EOF:        GOTO       CMDLBL(EXIT)
ERROR: /*************** ERROR HANDLING ROUTINE ***************/
            IF         COND(&ERROR) THEN(GOTO CMDLBL(EXIT))
            CHGVAR     VAR(&ERROR) VALUE(′1′)
RECEIVE:    RCVMSG     MSGTYPE(*ANY) MSGDTA(&MSGDTA) MSGID(&MSGID) +
                         RTNTYPE(&RTNTYPE) MSGF(&MSGF) +
                         MSGFLIB(&MSGLIB)
            IF         COND(&RTNTYPE *NE ′15′) THEN(DO) /* NOT +
                         EXCAPE MESSAGE    */
            SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                         MSGTYPE(*DIAG)
            GOTO       CMDLBL(RECEIVE)
            ENDDO
            SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                         MSGTYPE(*ESCAPE)
EXIT:       CHGVAR     VAR(&TOTAL) VALUE(&TOTAL - &FAIL)
            CHGVAR     VAR(&TOTALC) VALUE(&TOTAL)
            IF         COND(&FAIL *NE 0) THEN(DO)
            CHGVAR     VAR(&FAILC) VALUE(&FAIL)
            SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) MSGDTA(&FAILC +
                         || ′ Objects not attached  ′ || &TOTALC +
                         || ′ Objects attached to authorization +
                         list′) MSGTYPE(*ESCAPE)
            ENDDO
            SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) MSGDTA(&TOTALC +
                         || ′ Objects attached to authorization +
                         list′) MSGTYPE(*COMP)
            ENDPGM
                         * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 5-8 (Part 2 of 2). Add Objects to Authorization Lists

5.12.3 Command Definitions

ALLAUTL -- Build List of Objects on Authorization Lists

```
  CRTCMD ALLAUTL  PGM(ALLAUTL1)


 SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
   100            CMD       PROMPT(′List All Objects on AUTL′)
   200            PARM      KWD(OUTFILE) TYPE(Q1) MIN(1) PROMPT(′Output +
   300                        File Name′)
   400  Q1:       QUAL      TYPE(*NAME) LEN(10)
   500            QUAL      TYPE(*NAME) LEN(10) DFT(*CURLIB) +
   600                        SPCVAL((*CURLIB) (*LIBL)) PROMPT(′Library′)
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 5-9. Build List of Objects on Authorization Lists

FIXAUTL -- Attach Objects to Authorization Lists

```
  CRTCMD FIXAUTL  PGM(FIXAUTL1)


 SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
   100            CMD       PROMPT(′Fix Authorization List′)
   200            PARM      KWD(FILE) TYPE(Q1) MIN(1) PROMPT(′File′)
   300  Q1:       QUAL      TYPE(*NAME) LEN(10)
   400            QUAL      TYPE(*NAME) LEN(10) DFT(*LIBL) +
   500                        SPCVAL((*CURLIB) (*LIBL)) PROMPT(′Library′)
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 5-10. Attach Objects to Authorization Lists

5.13 Summary

Authorization lists provide a convenient grouping function of objects and users. We have discussed many of the advantages of authorization lists:

- Securing an object can authorize all the users on an authorization list to a object in one operation.
- Adding a user to an authorization list authorizes that user to all objects secured by the authorization list.
- The restore of objects to the system where they were saved automatically attaches the object to an authorization list.
- Authorization lists reduce the number of authority entries and the time to perform system back up is reduced.

OS/400 provides you many functions to set up AS/400 security environment, for example, authorization list, group profile, adopted authority, and so on. You should evaluate these functions from the policies and strategies of your company to decide which is the key point, for example, performance, save/restore operation, security severity, and so on. But the most important is **don′t let your AS/400 security environment get too complicated. Keep it simple and easy to maintain**.

Chapter 6. Audit Journal

OS/400 Version 2 Release 3 (V2R3) added significant new auditing features. Prior to V2R3 the auditing was limited to system wide audit of user actions. This release extends the audit function to include:

- System wide audit
  - Action auditing
  - Object access auditing
- Specific objects
  - All users′ access
  - Selected user′s access
- Selected Users
  - Additional action auditing
  - Audit of selected objects

Note: The auditing features provided by the audit journal can be used by any installation. You **do not** need system value QSECURITY set to 50 to use these features.

To take advantage of the OS/400 system-provided audit functions and thereby ensure that security is implemented at an appropriate level, the audit journal must be created and the audit criteria established in system values, user profiles and/or selected objects.

The following diagram provides an overview of the auditing structure. There are five system values related to auditing, two CL commands, and three parameters in the user profile.

```
┌───────────┐     ┌─────────┐     ┌───────────┐
│  System   │     │  Audit  │     │   User    │
│  Values   │     │  Events │     │  Profile  │
└───────────┘     └─────────┘     └───────────┘
 QAUDCTL           Action:         Values:
 QAUDLVL            CHGUSRAUD       OBJAUD
 QAUDENDACN        Object:          AUDLVL
 QAUDFRCLVL         CHGOBJAUD       *AUDIT SPCAUT
 QCRTOBJAUD
```

Figure 6-1. Overview of OS/400 Security Auditing Structure

AS/400 Security Reference has detailed information about how to set up and use the Audit Journal as well as the files used.

6.1 Audit Journal Events

Figure 6-2 gives an overview of the steps necessary to produce a report on given auditable events. Details are discussed in the following sections. The numbers in the figure are keys to the notes following the figure.

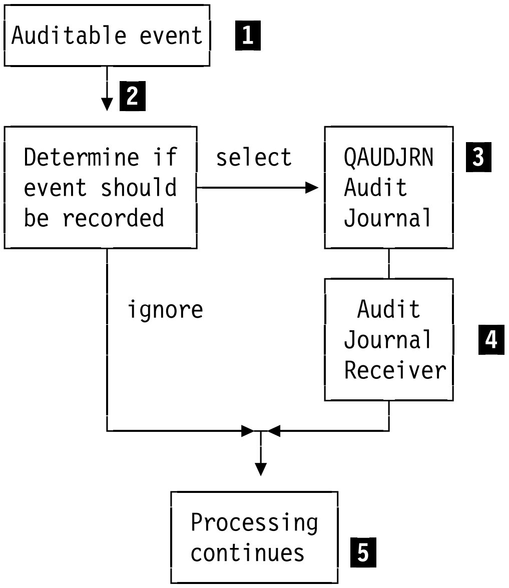


Figure 6-2. Audit Journal Events

Notes:

1 A user or program performs some activity that triggers an auditable event.

2 The system determines whether the event should be recorded. Events not recorded are simply discarded.

3 The journal QAUDJRN identifies the journal receiver where the event is recorded.

4 The data for selected events is recorded in the journal receiver. The event data describes includes:

- Date/Time
- Job name
- Event type
- User profile
- Object name
- Variable data based on specific event

5 The application program continues processing unaware of the fact that the event was recorded or discarded.

6.2 Creating the QAUDJRN Journal

Refer to the *AS/400 Security Reference* to find how to create the QAUDJRN journal.

6.3 System Values That Control Auditing

These system values control auditing on the system:

QAUDCTL	Auditing control
QAUDENDACN	Auditing end action
QAUDFRCLVL	Auditing force level
QAUDLVL	Auditing level
QCRTOBJAUD	Auditing for New Objects

Descriptions of these system values follow.

6.3.1 Auditing Control (QAUDCTL)

The QAUDCTL system value determines whether auditing is performed. It functions like an *on* and *off* switch.

6.3.2 Auditing End Action (QAUDENDACN)

The QAUDENDACN system value determines what action the system takes if auditing is active and the system is unable to write entries to the audit journal.

6.3.3 Auditing Force Level (QAUDFRCLVL)

The QAUDFRCLVL system value determines how often new audit journal entries are forced from memory to auxiliary storage. This system value controls the amount of auditing data that may be lost if the system ends abnormally.

6.3.4 Auditing Level (QAUDLVL)

The QAUDLVL system value determines which security-related events are logged to the security audit journal (QAUDJRN) for all users on the system. You can specify more than one value for the QAUDLVL system value, unless you specify *NONE.

Action Auditing

For a complete description of all the **Action Auditing Values** available on the QAUDLVL system value and the CHGUSRAUD command, refer to the *AS/400 Security Reference*.

Recommended value: The security-related events that are logged to the QAUDJRN journal depend on the setting of the system value QAUDLVL. We recommend that you specify one or more of the following values depending on your requirements:

****NONE:*** No security related events are logged (If *NONE is specified, no other value can be specified) - no system-wide auditing is performed.

****AUTFAIL:*** The system logs a journal entry (type AF or PW) for each authorization failure.

****SECURITY:*** System logs a journal entry (type CA, CP, SV, DS, JD, NA, OW, PA, or SE) for each security related function, including:

- Changing object authority
- Creating, changing, and restoring user profiles
- Changing system values

- Resetting the DST password
- Specifying a user profile name in the job description
- Changing network attributes
- Changing object ownership
- Changing programs to adopt the owner′s authority
- Changing subsystem routing

DELETE: The system logs a journal entry (type DO) for each delete operation (no journal entry is written for objects deleted from library QTEMP).

SAVRST: The system logs a journal entry (type RA, RJ, RO, RP, or RU) for each restore operation that modifies security, including:

- Changes to authority of objects
- Job descriptions that contain user names
- Objects with ownership changes
- Programs that run with adopted authority
- Authority changes to user profiles

Note: There are other values that can be specified with this systam value including: *CREATE, *JOBDTA, *OBJMGT, *OFCSRV, *PGMADP, *PGMFAIL, *PRTDTA, *SERVICE, *SPLFDTA, and *SYSMGT. These may also need to be changed depending on your auditing requirements.

A change to this system value takes effect immediately for all jobs running in the system.

To log an entry for every security-related event to the QAUDJRN journal change the system value QAUDLVL with the following command and parameter values:

```
CHGSYSVAL SYSVAL(QAUDLVL) VALUE('*AUTFAIL *SAVRST *DELETE *SECURITY')
```

User Auditing and the Change User Auditing (CHGUSRAUD) command are covered in 3.2, “User Auditing” on page 3-6.

6.3.5 Auditing for New Objects (QCRTOBJAUD)

Once the object auditing environment has been setup, only new objects require to have their auditing characteristics set such that their use can be audited. This can be done easily with the QCRTOBJAUD system value, which is used to specify the auditing for a new object if the auditing default for the library of the new object is set to *SYSVAL.

6.4 Using the Audit Journal to Report on System Activity

Based on entries in the audit journal receivers, the auditor can produce various reports to monitor security related system activity. Depending on the QAUDLVL system values, different journal entry types are logged, and can by various means be displayed or listed. Examples are provided in the following sections.

6.4.1 QAUDJRN Journal

The audit journal QAUDJRN in library QSYS is processed like any other journal. Files should not be journaled to it. We recommend that you keep it solely for security information.

System entries that appear in this journal are entries identified by a journal code of J. These entries relate to initial program load (IPL), and general operations performed on journal receivers (for example, a save of the receiver). The security related audit journal entries have a journal code of T.

Recovery from a damaged audit journal or audit journal receiver is the same as for other journals. See the *AS/400 Advanced Backup and Recovery Guide* for recovery information.

6.4.2 Audit Journal Flow

Figure 6-3 gives an overview of the steps necessary to produce a report on given auditable events. Details are discussed in the following sections. The numbers in the figure are keyed to the notes following the figure.

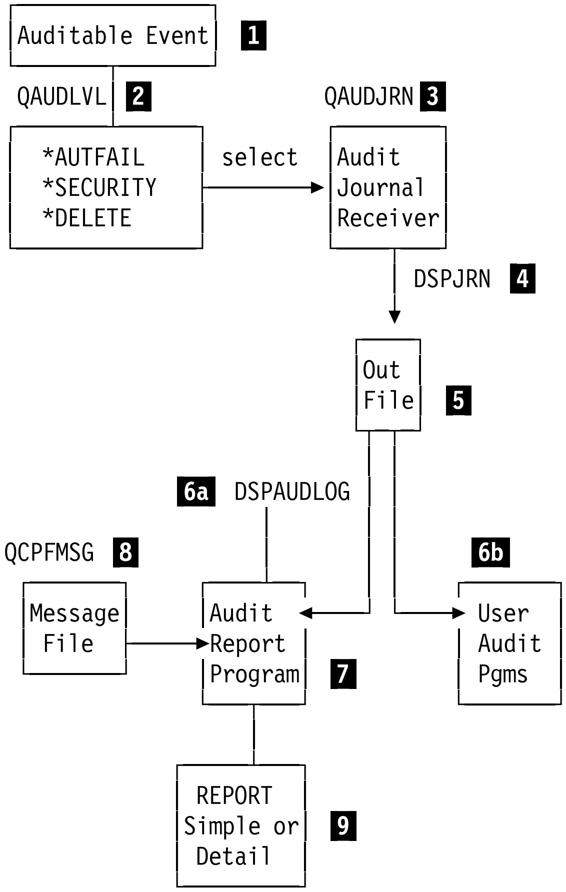


Figure 6-3. Audit Journal Flow

Notes:

1 A user or program performs some activity that triggers an auditable event.

2 The system value QAUDLVL determines whether the event should be audited (in this case *AUTFAIL, *SECURITY, and *DELETE).

3 The journal QAUDJRN identifies the journal receiver where the event is recorded.

4 The DSPJRN command can be used to select the events that are to be written to a database file (OUTFILE). The selection criteria include:

- Date/Time
- Job name
- Event types
- User profile

5 The database file created contains information that can be processed by application programs. This output is optimized for program access (Query/400, for example) and is not designed to be viewed on the screen with the DSPJRN command.

6a The DSPAUDLOG tool in the QUSRTOOL library can be used to interpret the information and print descriptive text for entries in the QAUDJRN journal. Output from the DSPAUDLOG command can be directed to a workstation or to a printer.

6b User-written audit programs can read, analyze, and summarize the audit data collected.

7 The program reads the output file and uses the event identifier to retrieve a message.

8 Messages in the QCPFMSG message file correspond to the events found in the audit log. The messages are defined so that the message data describes the fields found in the events.

9 The report can be a single line per event or a detailed report that provides details on the cause of the event.

6.4.3 Journal Entry Types for QAUDJRN

Each security audit journal entry contains the standard prefix fields for any journal entry (for example, date, time, journal sequence number).

6.4.4 Completeness of Audit Journal Entries

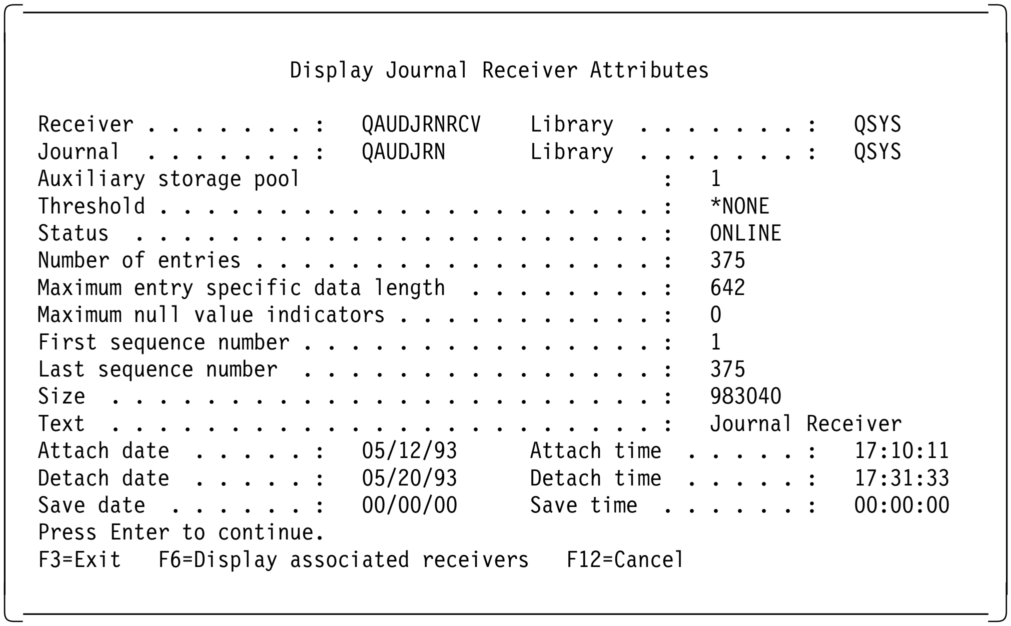
It is important to verify whether the journal was active for the entire period that you are reviewing. Intervals when the journal was not active should be identified. If such a situation exists, consider the impact on the overall evaluation. The system assigns a sequence number to each journal entry made. When a receiver is full, a new receiver has to be attached.

To identify all the receivers used for QAUDJRN, use the command:

```
WRKJRNA JRN(QAUDJRN)
```

and work with the receiver directory (F15). Display the attributes of the receiver and note the attach/detach dates and times, as well as the first and last sequence numbers. See the example in Figure 6-4. The detach time and date of the first receiver should equal the attach time and date of the second receiver. Continue this comparison through all the receivers on the directory for the time period you are concerned with.

Note: Individual journal entries cannot be deleted; however, the entire receiver can be.



```
                    Display Journal Receiver Attributes

Receiver . . . . . . . :   QAUDJRNRCV    Library . . . . . . . :   QSYS
Journal  . . . . . . . :   QAUDJRN       Library . . . . . . . :   QSYS
Auxiliary storage pool                              :   1
Threshold . . . . . . . . . . . . . . . . . . . . . :   *NONE
Status  . . . . . . . . . . . . . . . . . . . . . . :   ONLINE
Number of entries . . . . . . . . . . . . . . . . . :   375
Maximum entry specific data length  . . . . . . . . :   642
Maximum null value indicators . . . . . . . . . . . :   0
First sequence number . . . . . . . . . . . . . . . :   1
Last sequence number  . . . . . . . . . . . . . . . :   375
Size  . . . . . . . . . . . . . . . . . . . . . . . :   983040
Text  . . . . . . . . . . . . . . . . . . . . . . . :   Journal Receiver
Attach date  . . . . . :   05/12/93      Attach time  . . . . . :   17:10:11
Detach date  . . . . . :   05/20/93      Detach time  . . . . . :   17:31:33
Save date  . . . . . . :   00/00/00      Save time  . . . . . . :   00:00:00
Press Enter to continue.
F3=Exit   F6=Display associated receivers   F12=Cancel
```

Figure 6-4. Receiver Attribute Panel

6.4.5 Converting Security Audit Journal Entries

You can use the outfile parameter on the Display Journal (DSPJRN) command to transfer the audit journal entries into a database file for future use. The OUTFILE parameter allows you to name a file or member. Depending on the OUTMBR parameter value, records can be added or replaced. If the file does not exist, a file is created using the record format QJORDJE2. This format, shown in Figure 6-5, defines the standard heading fields for each journal entry. The audit data sent to the outfile can then be analyzed by using Query/400, SQL/400, a user-written program, or the DSPAUDLOG tool. The DSPAUDLOG tool is described in the following section.

Offset	Field	Format	Description
1	Entry Length	Zoned(5,0)	Total length of the journal entry
6	Sequence Number	Zoned(10,0)	Applied to each journal entry
16	Journal Code	Char(1)	Always T
17	Entry Type	Char(2)	Security related event (see Table 15-9 on page 15-23)
19	Date of Entry	Char(6)	System date the entry was made
25	Time of Entry	Zoned(6,0)	System time the entry was made
31	Name of Job	Char(10)	Name of the job that caused the entry to be generated
41	Name of User	Char(10)	User profile name associated with the job
51	Job Number	Zoned(6,0)	The job number
57	Program Name	Char(10)	Name of the program that made the journal entry
67	Object Name	Char(10)	Name of the object
77	Library Name	Char(10)	Name of the library the object is in
87	Member Name	Char(10)	Name of the member
97	Count/RRN	Zoned(10,0)	Change of count or relative record number
107	Flag	Char(1)	Flag byte: 1 or 0
108	Commit Cycle ID	Char(10)	Commit cycle identifier
118	User Profile	Char(10)	Name of the current user profile
128	System Name	Char(8)	Name of the system
136	Reserved	Char(20)	Not used
156	Specific Data	Char(144)	Entry specific data

Figure 6-5. QJORDJE2 Record Format

The following examples illustrate the technique to produce reports from the audit journal entries.

Example 1: Report all security related events that happened within a specific time.

Step 1. Use DSPJRN to select all security related events logged:

```
DSPJRN  JRN(QSYS/QAUDJRN) FROMTIME('05/19/93' '08:00:00')
        TOTIME('05/19/93' '18:00:00') ENTTYP(*ALL)
        USRPRF(*ALL) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2)
        OUTFILE(SECURITY/AUDSECEV)
```

The parameter OUTFILFMT allows either *TYPE1 or *TYPE2. Specifying *TYPE2 ensures that the converted journal entries contain the user profile field. Specifying *TYPE1 converts the journal entries without the user profile field.

Step 2. The query definitions in Figure 6-6 applied to the AUDSECEV file (created above) produce the requested report, shown in Figure 6-7. The report is based on record format QJORDJE2.

```
5738QU1                          IBM AS/400 Query

  Query . . . . . . . . . . . . . . . . . AUDSECEV
    Library . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . EBCDIC
  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No  (default)
  Special conditions
    *** All records selected by default ***

Selected files
  ID     File           Library       Member        Record Format
  T01    AUDSECEV       SECURITY      *FIRST        QJORDJE2
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  JODATE                                       Date of entry
  JOTIME                                       Time of entry
  JOENTT                                       Entry Type
  JOJOB                                        Name of Job
  JONBR                                        Number of Job
  JOUSER                                       Name of User
  JOPGM                                        Name of Program
  JOUSPF                                       User Profile
Report column formatting and summary functions
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count


  Field           Summary    Column                              Dec
  Name            Functions  Spacing  Column Headings        Len  Pos
  JODATE                     0        DATE                     6
  JOTIME                     2        TIME                     6    0
  JOENTT                     2        TYPE                     2
  JOJOB                      2        JOB                     10
                                      NAME
  JONBR                      2        JOB                      6    0
                                      NUMBER
  JOUSER                     2        USER                    10
                                      NAME
  JOPGM                      2        PROGRAM                 10
                                      NAME
  JOUSPF                     2        USER                    10
* * *   E N D   O F   Q U E R Y   * * *
```

Figure 6-6. Query Selections to Report Security Related Events

The problem with this approach is that the 144-character ″Entry Specific Data″ field (as shown in Figure 6-5 on page 6-8), is just one field which contains details for each event logged. As different logged events record logged data in this field, it is not sufficient to include this single field in the Query report, as there are no headings or formatting to tell the user how to interpret the data.

```
                                    SECURITY RELATED EVENTS             PAGE    1
DATE        TIME  TYPE  JOB         JOB      USER        PROGRAM     USER
                        NAME        NUMBER   NAME        NAME        PROFILE
051993   8:51:06   SV   DSP19         3295   QSECOFR     QCMD        QSECOFR
051993  10:50:06   SV   SCPF             0   QSYS        QWCISCFR    QSYS
051993  10:50:06   SV   SCPF             0   QSYS        QWCISCFR    QSYS
051993  10:52:31   CA   QSYSARB       3302   QSYS        QWCAINAR    QSYS
051993  11:14:29   CA   DSP01         3305   QSECOFR     QCMD        QSECOFR
051993  11:14:30   CA   DSP01         3305   QSECOFR     QCMD        QSECOFR
051993  11:14:46   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:15:08   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:15:26   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:16:32   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:17:24   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:18:27   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:19:20   CA   DSP19         3319   QSECOFR     QCMD        QSECOFR
051993  11:19:53   PW   QINTER        3315   QSYS        QWTMCMNL    QSYS
051993  11:28:52   CA   DSP01         3305   QSECOFR     QQUSTRQRY   QSECOFR
051993  11:38:01   PW   QINTER        3315   QSYS        QWTMCMNL    QSYS
051993  11:57:51   CA   DSP19         3326   BERND       QQUSTRQRY   BERND
051993  12:06:29   CA   DSP19         3326   BERND       QQUSTRQRY   BERND
051993  12:06:37   CA   DSP19         3326   BERND       QQUSTRQRY   BERND
051993  13:00:29   PW   QINTER        3315   QSYS        QWTMCMNL    QSYS
051993  13:14:07   CA   DSP21         3335   JAY         QQUSTRQRY   JAY
051993  13:14:07   CA   DSP21         3335   JAY         QQUSTRQRY   JAY
051993  13:43:50   CA   SCPF             0   QSYS        QWCISCFR    QSYS
051993  13:43:50   CA   SCPF             0   QSYS        QWCISCFR    QSYS
051993  13:45:07   CA   DSP19         3336   BERND       QCMD        BERND
051993  13:45:11   CA   DSP19         3336   BERND       QCMD        BERND
051993  13:50:40   CA   DSP19         3336   BERND       QCMD        BERND
051993  15:47:19   PW   QINTER        3315   QSYS        QWTMCMNL    QSYS
051993  16:12:37   DO   DSP08         3380   SANTIAGO    QCMD        SANTIAGO
051993  16:53:06   CA   DSP08         3380   SANTIAGO    QCMD        SANTIAGO
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-7. Query Report of Security Related Events

To get around this problem, and extract usable audit data contained from offset 156 of record format QJORDJE2, we have to use another technique using the CRTDUPOBJ command. This creates an empty output file with data fields and headings defined to contain (and format) the data from the DSPJRN command. Each journal entry type has a database file that maps the entire journal entry. See the *AS/400 Security Reference* for description of the different output file names and the entry-specific data for the QAUDJRN journal.

Example 2: Report object authority changes made by QSECOFR (the CA journal code).

Step 1. Create a duplicate object of the physical file QASYCAJE using the CRTDUPOBJ command:

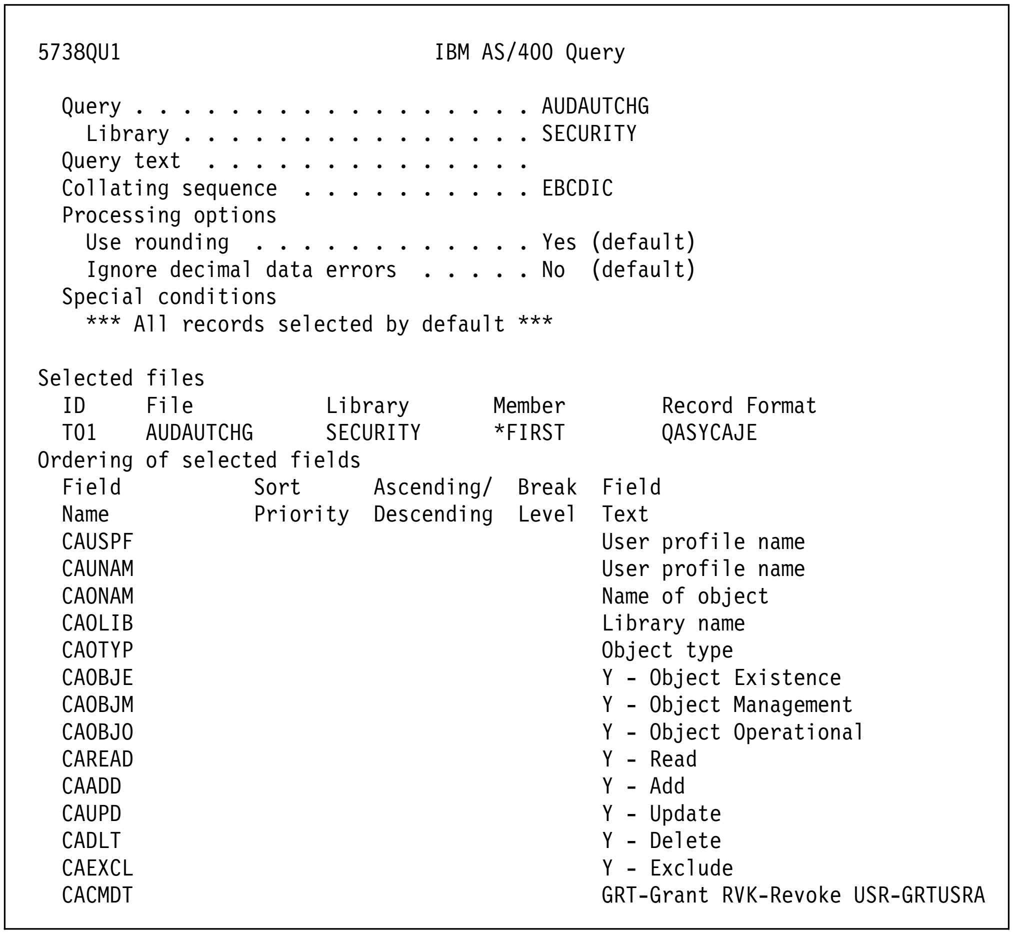
```
CRTDUPOBJ   OBJ(QASYCAJE) FROMLIB(QSYS) OBJTYPE(*FILE)
            TOLIB(SECURITY) NEWOBJ(AUDAUTCHG)
```

Step 2. Use the DSPJRN command shown below to move detailed information of authority changes to the database file AUDAUTCHG:

```
DSPJRN  JRN(QSYS/QAUDJRN) FROMTIME('05/13/93' '08:00:00')
TOTIME('05/13/93' '18:00:00') ENTTYP(CA) USRPRF(QSECOFR)
OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2) OUTFILE(SECURITY/AUDAUTCHG)
```

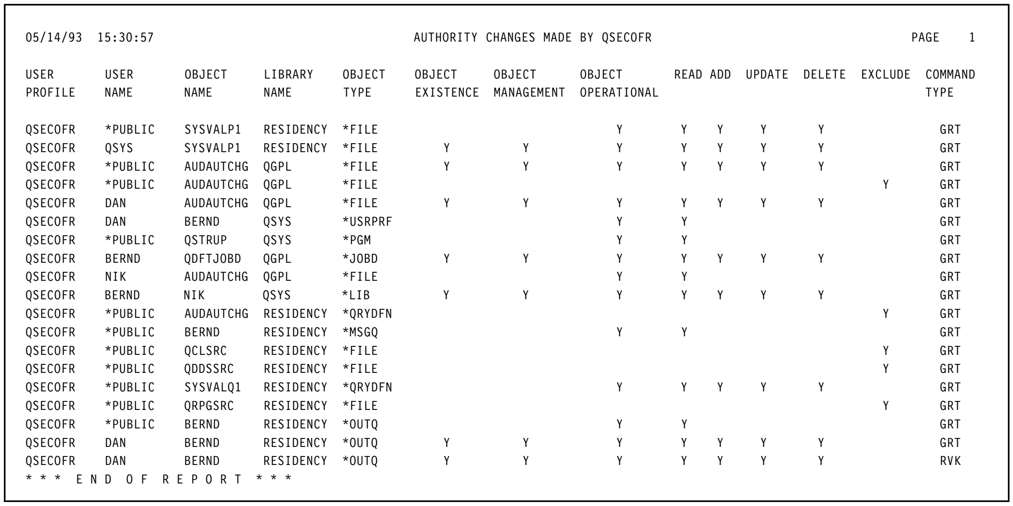
Note: The entry data length (parameter ENTDTALEN) defaults to the record length of the file created by the CRTDUPOBJ command.

Step 3. Use the query shown in Figure 6-8 to obtain the requested report shown in Figure 6-9. Note how the AUDAUTCHG file provides the field detail and column headings in the report.



```
5738QU1                          IBM AS/400 Query

  Query . . . . . . . . . . . . . . . . . AUDAUTCHG
    Library . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . EBCDIC
  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No  (default)
  Special conditions
    *** All records selected by default ***

Selected files
  ID     File           Library       Member        Record Format
  T01    AUDAUTCHG      SECURITY      *FIRST        QASYCAJE
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  CAUSPF                                       User profile name
  CAUNAM                                       User profile name
  CAONAM                                       Name of object
  CAOLIB                                       Library name
  CAOTYP                                       Object type
  CAOBJE                                       Y - Object Existence
  CAOBJM                                       Y - Object Management
  CAOBJO                                       Y - Object Operational
  CAREAD                                       Y - Read
  CAADD                                        Y - Add
  CAUPD                                        Y - Update
  CADLT                                        Y - Delete
  CAEXCL                                       Y - Exclude
  CACMDT                                       GRT-Grant RVK-Revoke USR-GRTUSRA
```

Figure 6-8. Query Selections to Report Authority Changes Made by QSECOFR



```
05/14/93  15:30:57                                    AUTHORITY CHANGES MADE BY QSECOFR                                    PAGE    1

USER       USER       OBJECT     LIBRARY    OBJECT    OBJECT     OBJECT      OBJECT       READ ADD  UPDATE  DELETE  EXCLUDE  COMMAND
PROFILE    NAME       NAME       NAME       TYPE      EXISTENCE  MANAGEMENT  OPERATIONAL                                     TYPE

QSECOFR    *PUBLIC    SYSVALP1   RESIDENCY  *FILE                               Y        Y    Y     Y       Y                GRT
QSECOFR    QSYS       SYSVALP1   RESIDENCY  *FILE         Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    *PUBLIC    AUDAUTCHG  QGPL       *FILE         Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    *PUBLIC    AUDAUTCHG  QGPL       *FILE                                                                    Y       GRT
QSECOFR    DAN        AUDAUTCHG  QGPL       *FILE         Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    DAN        BERND      QSYS       *USRPRF                               Y        Y                                   GRT
QSECOFR    *PUBLIC    QSTRUP     QSYS       *PGM                                  Y        Y                                   GRT
QSECOFR    BERND      QDFTJOBD   QGPL       *JOBD         Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    NIK        AUDAUTCHG  QGPL       *FILE                                 Y        Y                                   GRT
QSECOFR    BERND      NIK        QSYS       *LIB          Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    *PUBLIC    AUDAUTCHG  RESIDENCY  *QRYDFN                                                                  Y       GRT
QSECOFR    *PUBLIC    BERND      RESIDENCY  *MSGQ                                 Y        Y                                   GRT
QSECOFR    *PUBLIC    QCLSRC     RESIDENCY  *FILE                                                                    Y       GRT
QSECOFR    *PUBLIC    QDDSSRC    RESIDENCY  *FILE                                                                    Y       GRT
QSECOFR    *PUBLIC    SYSVALQ1   RESIDENCY  *QRYDFN                               Y        Y    Y     Y       Y                GRT
QSECOFR    *PUBLIC    QRPGSRC    RESIDENCY  *FILE                                                                    Y       GRT
QSECOFR    *PUBLIC    BERND      RESIDENCY  *OUTQ                                 Y        Y                                   GRT
QSECOFR    DAN        BERND      RESIDENCY  *OUTQ         Y          Y            Y        Y    Y     Y       Y                GRT
QSECOFR    DAN        BERND      RESIDENCY  *OUTQ         Y          Y            Y        Y    Y     Y       Y                RVK
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-9. Detailed Query Report of Authority Changes Made by QSECOFR

6.4.6 Display Audit Log Command

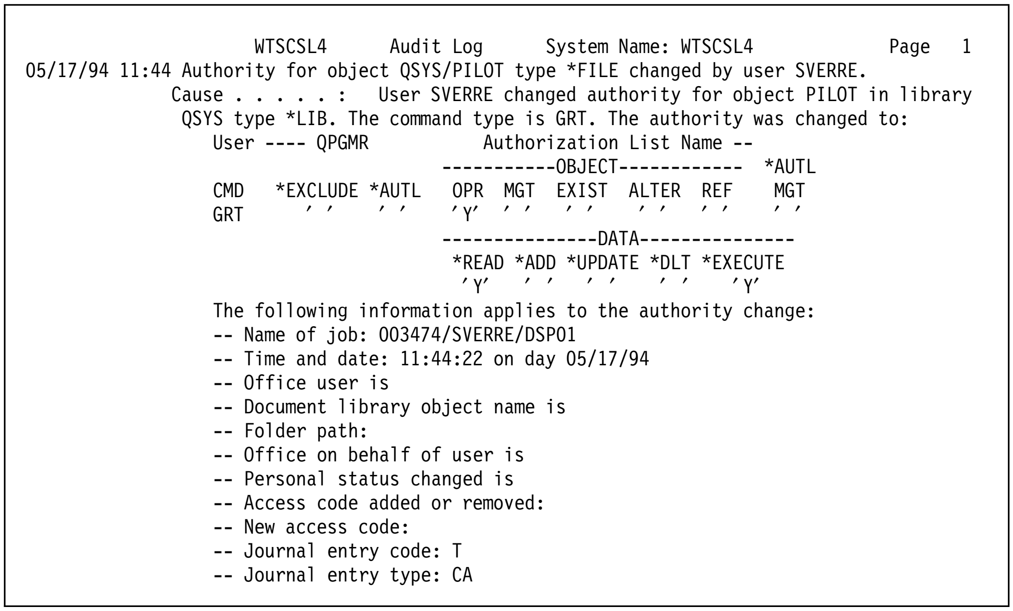
The Display Audit Log (DSPAUDLOG) command displays the contents of the security audit journal (QAUDJRN) by retrieving messages that make the entries in the audit journal readable by the user. The output is always printed to a spool file. The default will use DSPSPLF to display the printed output.

The DSPAUDLOG command source is provided in library QUSRTOOL. Note that QUSRTOOL contains only examples provided on an *as is* basis. QUSRTOOL programs need to be created before they can be used. If you issue the DSPAUDLOG command and you receive an error message, the DSPAUDLOG command may either not be created, or not be in your library list. For more information about the QUSRTOOL library and how to create the tools refer to the member AAAMAP in the QUSRTOOL/QATTINFO file. The member DSPAUDLOG in QUSRTOOL/QATTINFO file has a complete description of the DSPAUDLOG command.

When using command DSPAUDLOG, you should first use the DSPJRN command to filter the data written to the outfile as shown with the display QSECOFR authority changes example on page 6-10. Then use command:

```
DSPAUDLOG OPTION(user-lib/outfile)
```

Place the cursor on the individual message you receive and press the "help" key for message details. Refer to Figure 6-10 for an example of a DSPAUDLOG report.



```
                      WTSCSL4      Audit Log      System Name: WTSCSL4            Page   1
05/17/94 11:44 Authority for object QSYS/PILOT type *FILE changed by user SVERRE.
              Cause . . . . . :   User SVERRE changed authority for object PILOT in library
               QSYS type *LIB. The command type is GRT. The authority was changed to:
                  User ---- QPGMR           Authorization List Name --
                                        -----------OBJECT------------  *AUTL
                  CMD   *EXCLUDE *AUTL   OPR  MGT  EXIST  ALTER  REF    MGT
                  GRT     ' '     ' '    'Y'   ' '  ' '   ' '   ' '     ' '
                                        ---------------DATA---------------
                                         *READ *ADD *UPDATE *DLT *EXECUTE
                                          'Y'  ' '   ' '    ' '    'Y'
                  The following information applies to the authority change:
                  -- Name of job: 003474/SVERRE/DSP01
                  -- Time and date: 11:44:22 on day 05/17/94
                  -- Office user is
                  -- Document library object name is
                  -- Folder path:
                  -- Office on behalf of user is
                  -- Personal status changed is
                  -- Access code added or removed:
                  -- New access code:
                  -- Journal entry code: T
                  -- Journal entry type: CA
```

Figure 6-10. DSPAUDLOG Report

6.4.7 Audit Reports by Journal Entry Type

The audit journal may record many different types of entries. All journal entry types have security implications. You should, as a minimum, review the following journal entries:

- AF - Object authority failure
- CA - Object authority change
- CP - User profile created or changed

- DO - Object deletions
- DS - DST Security Officer password reset
- OW - Object owner change
- PA - CHGPGM used to change to program adopt
- PW - Invalid password entered or invalid user profile entered
- SV - System value changed

Recommendation: Query/400 is recommended for analyzing the audit data. It is more flexible than the DSPAUDLOG command and, for general users, it is much easier to use than programming in a high-level language. After using the DSPJRN command to filter a specific journal entry type to an outfile, use the Start Query (STRQRY) command to reach the Query menus. The fields will already be defined for you. Query can be used to select, sort, and summarize fields, which can be very useful in the analysis of the data. Refer to the *AS/400 Security Reference* for the layout of the records in the journal entry file you have copied.

In the following sections, the journal entry types mentioned previously are described, and some report examples are given.

Authority Failure (AF)

If the system value QAUDLVL *AUTFAIL is activated, every unsuccessful attempt to sign on or to access an object is recorded as a violation with an entry type "AF." The log can be used in several ways. If you are interested in which sensitive objects experienced unsuccessful access attempts, sort and summarize your report by object. If you are interested in which user profiles are causing the most violations, sort and count by user profile. See Figure 6-11 for the query definition and Figure 6-12 for the report produced.

A pattern of too many violations may indicate that your system is under attack, too "few" violations may indicate that security logging could be more inclusive.

Scan the "violation type" listed in your report. An "A" indicates an authorization violation, for example, someone tried to access an object for which they did not have authority.

The violation types can also be used to assess whether programs on your system are using restricted machine interface (MI) code that can potentially cause security breaches under level 30 or less. A "B" for blocked instruction or a "D" for domain violation may indicate the presence of restricted MI calls in some of the programs running on your system. Level 40 and 50 will not permit those programs to run.

Full or Sample Audits

When choosing to perform auditing on your system where security requirements are not very high, you may decide not to audit the system all of the time. Instead, an appropriate approach is to perform a sample audit, on a semi-regular basis, such as once a day for 2 hours. If this is run at random times each day, then it will offer some indication of security-related activity, without the system performance and disk space overhead.

```
Selected files
  ID     File           Library      Member       Record Format
  T01    JAYAUDAF       SECURITY     JAYAUDAF     QASYAFJE
Ordering of selected fields
  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text
  AFUSPF         10        A           1      User profile name
  AFVIOL                                      Violation type
  AFONAM                                      Name of object
  AFOLIB                                      Library name
  AFOTYP                                      Object type
  AFDATE         20        D                  Date of entry
  AFTIME                                      Time of entry
  AFPGM                                       Name of program
Report column formatting and summary functions
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count


                                                                             _________________
  Field          Summary    Column                                 Dec            Dec  Numeric
  Name           Functions  Spacing  Column Headings         Len  Pos       Len  Pos  Editing
  AFUSPF         5          0        USER                     10
                                     PROFILE
  AFVIOL                    2        VIOLATION                 1
                                     TYPE
  AFONAM                    2        OBJECT                   10
                                     NAME
  AFOLIB                    2        LIBRARY                  10
                                     NAME
```

Figure 6-11. Query Definition for Authority Failure

```
05/19/93  16:33:57                List of Authorization Violations

                                         May 1993
      USER       VIOLATION  OBJECT      LIBRARY     OBJECT    DATE        TIME  PROGRAM
      PROFILE    TYPE       NAME        NAME        TYPE                        NAME
      BERND          A      QAUDJRN     QSYS        *JRN      050993  10-53-33  QCMD
                     A      WRKJRN      QSYS        *CMD      050993  10-56-41  QCMD
                     A      QAUDJRN     QSYS        *JRN      050993  10-59-20  QCMD
                     A      JAY         QSYS        *USRPRF   050993  11-02-28  QCMD
                     A      QAUDJRN     QSYS        *JRN      050893   8-31-40  QCMD
                     A      QCLSRC      QGPL        *FILE     050893  13-55-43  QCMD
                     A      QATTINFO    QUSRTOOL    *FILE     050793  10-02-00  QCMD
                     A      QAUDJRN     QSYS        *JRN      050793  10-47-17  QCMD
                     A      QAUDJRN     QSYS        *JRN      050793  10-51-32  QCMD
                     A      QASYAFJE    RESIDENCY   *FILE     050793  11-00-22  QCMD
COUNT 10
      DAN            A      QSECOFR     QSYS        *USRPRF   050993   9-35-54  QCMD
                     A      DECISION    QSYS        *MENU     050793   8-33-24  QCMD
* * *   E N D   O F   R E P O R T   * * *
```

Figure 6-12. Authority Failure Report

Password and User Profile (PW)

Password and user profile violations are logged with entry type “PW” if QAUDLVL *AUTFAIL is activated.

Figure 6-13 shows a report on this entry type. Multiple invalid signons may be an indication that an unauthorized user has gained knowledge of a valid user profile and is trying to guess the password to break into the system.

The report can also tell you what physical device was used. Violation type “P” indicates a password violation, while violation type “U” indicates that the user ID is not valid.

```
05/19/93  18:00:55                                                      PAGE    1
                                        Invalid Password Attempts
                                             May 1993
USER        VIOLATION  DATE        TIME  TYPE  SYSTEM    DEVICE
NAME        TYPE                               NAME      NAME
BERND       P          050993  17:00:04   PW   WTSCSL4   DSP12
            P          050993  17:00:02   PW   WTSCSL4   DSP12
            P          050993  16:38:49   PW   WTSCSL4   DSP16
      COUNT 3
DAN         P          050793  14:23:09   PW   WTSCSL4   QWERTYS1
      COUNT 1
JR          P          051293  13:14:28   PW   WTSCSL4   DSP12
      COUNT 1
NIK         P          050993  10:51:05   PW   WTSCSL4   QWERTY00
            P          050993  10:51:01   PW   WTSCSL4   QWERTY00
            P          050993  10:50:56   PW   WTSCSL4   QWERTY00
            P          050993  10:50:48   PW   WTSCSL4   QWERTY00
            P          050993  10:50:24   PW   WTSCSL4   QWERTY00
* * *   E N D   O F   R E P O R T   * * *
```

Figure 6-13. Invalid Password Attempts Report

Authority Changes (CA)

When QAUDLVL *SECURITY is activated, every authority change is recorded. A report on authority changes, entry type "CA," for each user profile can therefore be produced. An example is provided in Figure 6-14, and also in Section 6.4.5, "Converting Security Audit Journal Entries" on page 6-7.

From this journal you also can monitor access changes to any sensitive files, programs, and other objects.

```
05/19/93  16:42:31                                                                                                     PAGE    1
                                                                 Authority Changes by Object
                                                                           May 1993
      OBJECT      LIBRARY     OBJECT    USER        OBJECT     OBJECT      OBJECT        READ  ADD  UPDATE  DELETE  EXCLUDE  COMMAND
      NAME        NAME        TYPE      NAME        EXISTENCE  MANAGEMENT  OPERATIONAL                                       TYPE
      AUDAFJENIK  NIK         *FILE     *PUBLIC                                                                     Y         GRT
                  NIK         *FILE     QSECOFR         Y          Y            Y          Y    Y     Y       Y       Y         GRT
COUNT 2
      AUDJRNCA1   RESIDENCY   *FILE     *PUBLIC                                 Y          Y    Y     Y       Y                 GRT
                  RESIDENCY   *FILE     QSYS            Y          Y            Y          Y    Y     Y       Y                 GRT
COUNT 2
      NIKTEST1    NIK         *QRYDFN   *PUBLIC                                 Y          Y    Y     Y       Y                 GRT
COUNT 1
      NIKTEST2    NIK         *QRYDFN   *PUBLIC                                 Y          Y    Y     Y       Y                 GRT
COUNT 1
FINAL TOTALS
COUNT 6
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-14. Authority Changes Report

Changes to User Profiles (CP)

Entry type "CP" indicates that security related changes have been made to a user profile. The date and time of the profile change as well as the user making the change are logged. This information can be used to review the work of the security officer or a security administrator. Constantly changing profiles may be questioned.

From this report you also can identify those users for which initial menu or program and limited capability parameters were changed. See the user profile change report in Figure 6-15.

```
05/19/93  16:45:17                                                                                       PAGE    1
                                                    User Profiles Changed
                                                              May 1993
USRPRF      USER        GROUP       GROUP       LIMIT         DATE        TIME  PASSWORD  *ALLOBJ  *SERVICE  *SECADM
NAME        PROFILE     PROFILE     AUTHORITY   CAPABILITIES                    CHANGED   SPCAUT   SPCAUT    SPCAUT
GROUP       DAN                     *NONE        *NO          051393  10:36:15     Y
            DAN                     *NONE        *NO          051393  10:47:47     Y
GROUPTEST1  DAN         GROUP       *NONE        *NO          051393  10:48:21     Y
GROUPTEST2  DAN         GROUP       *NONE        *NO          051393  10:48:30     Y
NIKTEST1    QSECOFR                 *NONE        *YES         051393  11:02:19     Y
PP          SANTIAGO                *NONE        *NO          051393  10:59:05     Y
SANTIAGO    SANTIAGO                                          051393  11:03:27
TESTUSER1   DAN         TEXT        *NONE        *NO          051393  10:45:41     Y
TESTUSER2   DAN         TEXT        *NONE        *NO          051393  10:45:51     Y
TESTUSER3   DAN         TEXT        *NONE        *NO          051393  10:45:59     Y
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-15. User Profile Changes Report

System Value Change (SV)

QAUDLVL with *SECURITY records changes to all security system values. These values should be reviewed to ensure that effective global security values have been established. The report indicates when the change occurred, who changed it, and the old and new values. An example of a system value change report is given in Figure 6-16.

From the report you can check whether there have been any changes to the present security level or other system values for a specific time period. The current system values on the system should also be compared to the recommended system values in 2.9, “System Values and Network Attributes Recommendations” on page 2-14. Significant deviations from the recommended values may seriously weaken overall security.

```
05/19/93  16:43:33                                                                                  PAGE    1
                                                        System Value Changes
                                                               May 1993
USER        DATE          TIME  SYSTEM      NEW                                         OLD
PROFILE                         VALUE       VALUE                                       VALUE
JAY         051493   9:08:04  QINACTITV   0000000060                                  *NONE
QSECOFR     050693  17:10:20  QAUDLVL     *AUTFAIL  *SECURITY *SAVRST   *DELETE       *NONE
            050693  16:55:02  QTIME       165500                                      175600
            050893  15:15:13  QPWDEXPITV  000044                                      *NOMAX
            050893  15:16:58  QPWDEXPITV  000360                                      000044
            050893  15:18:03  QPWDEXPITV  *NOMAX                                      000360
            050893  15:19:39  QPWDMAXLEN  0000000008                                  0000000010
            050893  15:19:49  QPWDMAXLEN  0000000010                                  0000000008
            050893  16:37:20  QRMTSIGN    *REJECT                                     *SAMEPRF
            050893  16:44:54  QRMTSIGN    *FRCSIGNON                                  *REJECT
            050993   8:51:51  QRMTSIGN    *SAMEPRF                                    *FRCSIGNON
            050993   8:56:05  QRMTSIGN    *VERIFY                                     *SAMEPRF
            051693  11:50:39  QLMTDEVSSN  1                                           0
            051693  11:53:43  QLMTDEVSSN  0                                           1
            051993   8:51:06  QSECURITY   40                                          30
QSYS        051293   8:05:06  QMCHPOOL    0000005619                                  0000005604
            051293   8:05:06  QBASACTLVL  0000000003                                  0000000003
            051493  19:48:53  QMCHPOOL    0000005759                                  0000005619
            051493  19:48:53  QBASACTLVL  0000000003                                  0000000003
            051993  10:50:06  QMCHPOOL    0000005759                                  0000005759
            051993  10:50:06  QBASACTLVL  0000000003                                  0000000003
SANTIAGO    050893  16:10:32  QPWDMINLEN  0000000002                                  0000000001
            050893  16:12:24  QPWDMINLEN  0000000001                                  0000000002
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-16. System Value Changes Report

Ownership Changes (OW)

This entry represents changes made to ownership. The old owner, the new owner, the name of the object, and the name of the library it resides in can be listed.

DST Password Reset Journal Entries (DS)

This journal entry records the setting of the Dedicated Service Tool (DST) password. DST is generally meant to be used by the IBM Service Engineer. During manual IPL, a password is needed to get into DST. The report will indicate whether the DST password is reset, and by which user profile. Requests to reset the DST password are also recorded.

Changed Programs that Adopt Authority (PA)

This entry type is logged when programs have been changed to adopt the owner′s authority. The name of the program, the library it is found in, and the name of the owner can be listed in a query report.

Deleting an Object (DO)

If QAUDLVL parameter *DELETE is specified, an entry is made in the journal when objects are deleted. It is therefore possible to monitor if, for example, sensitive objects that should not have been deleted, have been deleted. Objects deleted from QSYS or from a sensitive production library can be identified.

A review of a “Deleted Objects Report” can easily disclose whether objects that are deleted have been removed on a timely basis. For example, the security officer should delete user profiles immediately after an employee is terminated.

An example of a report for this purpose is shown in Figure 6-17.

```
05/21/93  09:41:29                                        PAGE    1
                       Deleted Objects By User Profile
                                    May 1993
      USER        OBJECT      LIBRARY     OBJECT    DATE        TIME
      PROFILE     NAME        NAME        TYPE
      BERND       AUTLP2      RESIDENCY   *FILE     052093  16:10:10
                  DISPLAYS    QSYS        *AUTL     052093  16:05:30
                  ADOPT       QSYS        *AUTL     052093  16:05:13
                  AUDAUTCHG   RESIDENCY   *FILE     051593  15:05:12
                  BWTEST1     RESIDENCY   *FILE     051593  11:20:15
                  BWTEST1     RESIDENCY   *FILE     051593  10:13:22
                  BWTEST1     RESIDENCY   *FILE     051593   9:56:18
                  BWTEST1     RESIDENCY   *FILE     051593   9:53:11
                  PRINTPRF    RESIDENCY   *FILE     051393  15:40:49
                  OBJAUTF     RESIDENCY   *FILE     051393   9:52:23
                  AUDTEST1    RESIDENCY   *FILE     050793  10:59:57
                  AUDTEST1    RESIDENCY   *FILE     050793  10:53:05
COUNT 12
      DAN         PWDCHKCL1   QSYS        *PGM      052193   8:36:23
                  BWTEST1     RESIDENCY   *FILE     051593  13:57:23
                  OBJAUTF     RESIDENCY   *FILE     051593  13:31:06
                  WTSCSL1     QSYS        *DEVD     051493  14:50:54
                  OBJAUTF     RESIDENCY   *FILE     051493  13:28:04
                  TESTUSER3   QSYS        *USRPRF   051493  11:22:10
                  TESTUSER2   QSYS        *USRPRF   051493  11:22:01
* * *   E N D   O F   R E P O R T   * * *
```

Figure 6-17. Deleted Objects Report

6.5 Specific User and Object Auditing

You can audit security related events at three levels:

- System-wide for all users
- Specific users
- Specific objects

The previous sections have described how you can implement system-wide auditing. The following section will explain how you can audit specific users as well as specific objects.

6.5.1 Specific User Action Auditing

Sometimes you may want to audit a specific user profile for more actions than other users are audited for, or you may want to monitor a specific user′s activities against a specific object.

Note!

The intention of auditing user access is to detect possible security violations. It is **NOT** to be misused. One example of misuse is to use it to watch the actions of a user.

In may countries such a misuse is prohibited by law. If ever in doubt seek legal advice.

AUDLVL on the User Profile

The AUDLVL parameter on the user profile defines the types of *actions* that are audited for a *specific* user profile.

AUDLVL is used in **in conjunction** with the system value QAUDLVL. If QAUDLVL is set to *DELETE and AUDLVL is set to *CREATE for a user profile, both *DELETE and *CREATE will be monitored. The AUDLVL parameter can be changed with the Change User Auditing (CHGUSRAUD) command.

Recommendation: You can use AUDLVL on the user profile if, for example, you want to audit or monitor all actions performed by users with *SECOFR user class or *ALLOBJ authority. See 6.5.4, “Auditing QSECOFR Activity” on page 6-20 for an example of implementing this environment.

OBJAUD on the User Profile

The OBJAUD parameter on the user profile specifies the *object* auditing value for the user. This value only takes effect if the object auditing (OBJAUD) value for the object being accessed has the value *USRPRF (see, Table 6-2 on page 6-19). The OBJAUD parameter can be changed with the Change User Auditing (CHGUSRAUD) and Change Document Library Object Auditing (CHGDLOAUD) commands. The possible options for OBJAUD are:

Table 6-1. Values for the OBJAUD Parameter on the User Profile

Value	Description
<u>*NONE</u>	The auditing value for the object determines when auditing is performed.
***CHANGE**	All change accesses by this user on all objects with the *USRPRF audit value are logged.
***ALL**	All change and read accesses by this user on all objects with *USRPRF audit value are logged.

6.5.2 Object Auditing

The object′s OBJAUD parameter identifies the type of auditing for an object. The valid values for the OBJAUD parameter can be changed with the CHGOBJAUD command. The possible values are:

Table 6-2. Values for the OBJAUD Parameter on the Object

Value	Description
<u>*NONE</u>	No auditing will occur for this object when it is read or changed regardless of the user who is accessing the object.
***USRPRF**	Audit this object only if the user accessing the object is being audited. The user profile for the job will be tested to determine if auditing should be done for this object. The user profile can specify if only change access will be audited or if both read and change accesses will be audited for this object.
***CHANGE**	Audit all change access to this object by all users on the system.
***ALL**	Audit all access to this object by all users on the system. All access is defined as a read or change operation.

Table 6-3 shows how the OBJAUD values for the user and the object work together:

Table 6-3. Auditing Performed for Object Access

OBJAUD value for object	**OBJAUD value for user**		
	*NONE	*CHANGE	*ALL
*NONE	None	None	None
*USRPRF	None	Change	Change and Read
*CHANGE	Change	Change	Change
*ALL	Change and Read	Change and Read	Change and Read

6.5.3 Document Library Object Auditing

The object′s AUDDLO parameter identifies the type of auditing for an object. The valid values for the AUDDLO parameter, which can be changed with the CHGDLOAUD command, are explained in Table 6-4 on page 6-20.

Table 6-4. Values for the AUDDLO Parameter on the Object

Value	Description
***SAME**	The level does not change.
***NONE**	No auditing will occur for this document or folder when it is read or changed regardless of the user who is accessing the object.
***USRPRF**	Audit this document or folder only if the user accessing the object is being audited. The user profile for the job will be tested to determine if auditing should be done for this object. The user profile can specify if only change access will be audited or if both read and change accesses will be audited for this object.
***CHANGE**	Audit all change access to this document or folder by all users on the system.
***ALL**	Audit all access to this document or folder by all users on the system. All access is defined as a read or change operation.

Table 6-5 shows how the OBJAUD values for the user and the AUDDLO for the object work together:

Table 6-5. Auditing Performed for Document Library Object Access

AUDDLO value for object	**OBJAUD value for user**		
	*NONE	*CHANGE	*ALL
*NONE	None	None	None
*USRPRF	None	Change	Change and Read
*CHANGE	Change	Change	Change
*ALL	Change and Read	Change and Read	Change and Read

6.5.4 Auditing QSECOFR Activity

Many AS/400 sites require activities of QSECOFR (or other *ALLOBJ users) to be monitored and recorded. The User Action Auditing capability offers a very easy way of implementing this. The following steps will guide you through setting up this environment:

1. Ensure that the QAUDJRN journal and a journal receiver have been created.
2. Ensure that the QAUDCTL system value has been set to include *AUDLVL.
3. Issue the command `CHGUSRAUD USRPRF(QSECOFR) AUDLVL(*CMD)`

After a period of activity, you can print a report of the commands issued by QSECOFR, using the following:

4. Copy the skeleton output file:

```
CRTDUPOBJ  OBJ(QASYCDJE) FROMLIB(QSYS) OBJTYPE(*FILE)
           TOLIB(SECURITY) NEWOBJ(SECOFRCMD)
```

5. Display the journal data into the created output file for the time/date range required:

```
DSPJRN JRN(QSYS/QAUDJRN) FROMTIME('05/19/93' '06:00:00')
       TOTIME('05/28/93' '18:00:00') ENTTYP(CD)
       USRPRF(QSECOFR) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2)
       OUTFILE(SECURITY/SECOFRCMD)
```

6. Run a Query program, such as in the following example, to produce the report shown.

```
5738QU1                            IBM Query/400
  Query . . . . . . . . . . . . . . . . . SECOFR
    Library . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . . Report on QSECOFR activity
  Collating sequence  . . . . . . . . . . Hexadecimal
  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes
  Special conditions
    *** All records selected by default ***

Selected files
  ID     File           Library       Member        Record Format
  T01    SECOFRCMD      SECURITY      *FIRST        QASYCDJE
Result fields
  Name        Expression                         Column Heading          Len  Dec
  DATE        substr(cddate,1,2) || ’/’ ||       Date
              substr(cddate,3,2) || ’/’ ||
              substr(cddate,5,2)
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  DATE
  CDTIME                                       Time of entry
  CDONAM                                       Name of object
  CDOLIB                                       Library name
  CDCLP                                        Y - Command run from CL pgm or REXX proc
  CDPGM                                        Name of program
  CDCMDS                                       Command string
Report column formatting and summary functions
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count           Overrides
  Field           Summary    Column                                 Dec  Null       Dec  Numeric
  Name            Functions  Spacing  Column Headings        Len  Pos  Cap   Len  Pos  Editing
  DATE                       0          Date                   8
  CDTIME                     1        Time                     6    0                     Yes
  CDONAM                     1        Object                  10
                                      Name
  CDOLIB                     1        Library                 10
                                       Name
  CDCLP                      0        CL                       1
                                      Pgm
  CDPGM                      1        Program                 10
                                      Name (if
                                      applicable)
  CDCMDS                     2        Command String        6000              70
```

Figure 6-18. Edited Query Definition for QSECOFR Command Report

```
                           QSECOFR Command Report                              PAGE    1
  Date       Time Object     Library   CL  Program      Command String
                  Name        Name     Pgm Name (if
                                           applicable)
06/20/93 18:40:35 WRKSYSVAL  QSYS       N  QCMD         WRKSYSVAL
06/20/93 18:40:43 WRKSYSSTS  QSYS       N  QCMD         WRKSYSSTS
06/20/93 18:49:11 WRKOBJPDM  QSYS       N  QCMD         WRKOBJPDM LIB(QCBL)
06/20/93 18:51:10 DSPLIB     QSYS       N  QCMD         DSPLIB LIB(TEST)
06/20/93 19:18:16 WRKOBJPDM  QSYS       N  QCMD         WRKOBJPDM LIB(EVANS)
06/20/93 19:19:32 ALCOBJ     QSYS       N  QCMD         ALCOBJ OBJ((QSYS/SECURITY *LIB *SHRRD))
06/20/93 19:19:38 DLCOBJ     QSYS       N  QCMD         DLCOBJ OBJ((QSYS/SECURITY *LIB *SHRRD))
06/20/93 19:20:57 CALL       QSYS       N  QCMD         CALL PGM(PAYROLL/DSPSALRY)
06/20/93 19:22:54 ADDLIBLE   QSYS       N  QCMD         ADDLIBLE LIB(SECURITY)
06/20/93 19:22:56 LOOKUP     SECURITY   N  QCMD         SECURITY/LOOKUP
06/20/93 19:22:56 RCVMSG     QSYS       Y  CHGDTAARA2   RCVMSG MSG(&MSG) RTNTYPE(&MSGTYPE)
06/20/93 19:22:56 SNDPGMMSG  QSYS       Y  CHGDTAARA2   SNDPGMMSG MSG('Validation value not correct')
06/20/93 19:23:33 FASTLOOK   SECURITY   N  QCMD         SECURITY/FASTLOOK DTAARA(WERWER)
* * *  E N D  O F  R E P O R T  * * *
```

Figure 6-19. Sample output for QSECOFR command Report

From this report you can investigate activity performed by QSECOFR such as running the Display Salary program from the Payroll application.

Chapter 7. Security Design For Performance

One of the causes of performance problems may be excessive security processing by the system. System performance on the AS/400 can change significantly based upon how objects are secured. This chapter provides an overview of the way AS/400 stores object authority and the authorization lookup process. This overview of the AS/400 authority checking internals gives you the background needed to understand the performance impact of different methods used to secure objects. Based on this understanding, the second part of the chapter gives practical rules you can use to secure data in the an efficient and effective manner.

7.1 Architecture

The hallmark of the AS/400 is the simplified system operation resulting from the integration of function by the system and operating system. Figure 7-1 contrasts the traditional system architecture (such as a host system) with the AS/400. Traditional systems often have separately installed products for communications, security, and data bases. The AS/400 integrates all of these features into OS/400. Access control of traditional systems often is done by a security product that is separate from the operating system. Traditional operating systems call this security product when access checks are needed. The security product performs the access control checks and returns a go or no-go indication. The operating system then proceeds or prevents the operation.

Like other AS/400 functions, security is highly integrated. All security functions are included in OS/400 and the AS/400 system. A separate security product is not required. In traditional systems often the different products have their own access control rules and enrollment. The AS/400 operating system, data base, and communications functions all use the same access check rules and security enrollment.

The AS/400 access control checks are built into the machine instructions. Each time an AS/400 program runs any instruction that uses a system pointer to reference data (an object), the machine interface makes an access control check as part of the instruction execution. If the user is authorized to the object, the instruction continues. If the user is not authorized, the instruction is not executed and the system signals a ″not authorized exception″. Moving security into the machine instructions prevents a system programmer from altering the operating system to bypass the access control checks. In traditional systems, a system programmer has the potential of altering the operating system to bypass security.

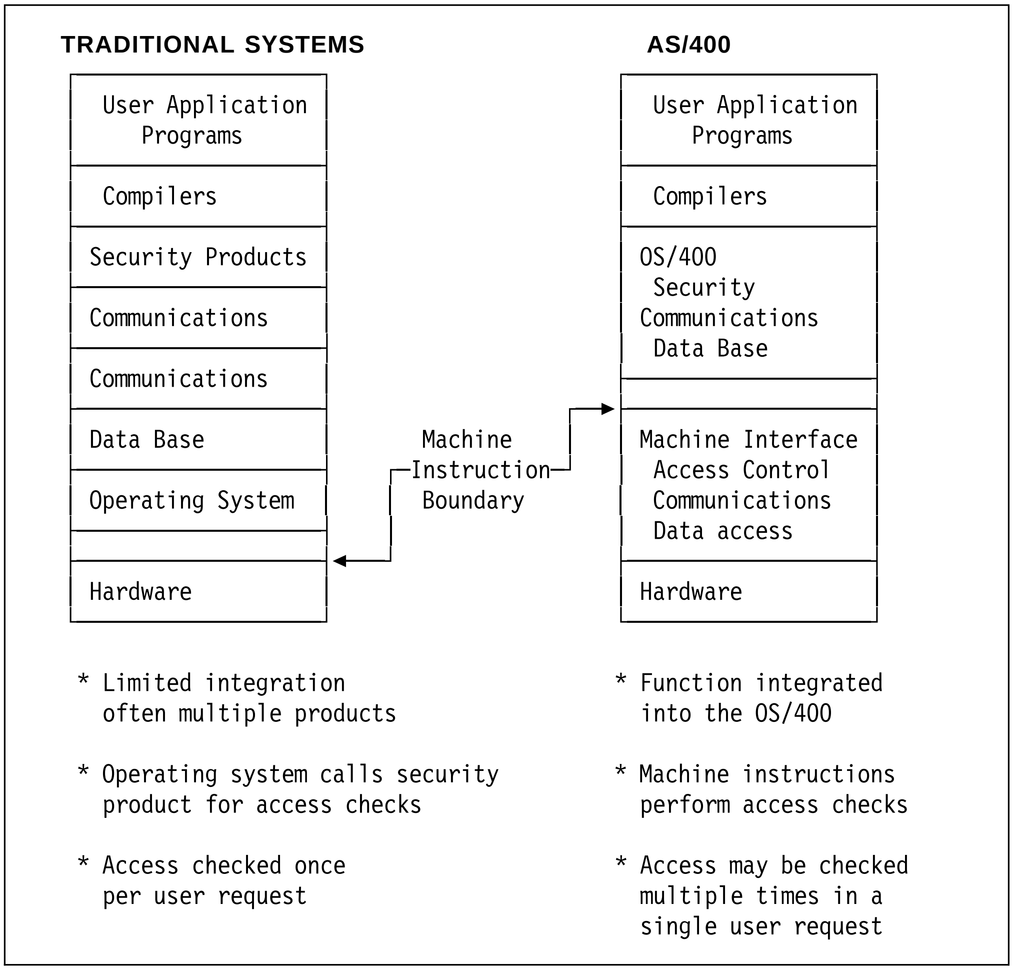

TRADITIONAL SYSTEMS
AS/400
User Application Programs
Compilers
Security Products
Communications
Communications
Data Base
Operating System
Hardware
User Application Programs
Compilers
OS/400
Security
Communications
Data Base
Machine Interface
Access Control
Communications
Data access
Hardware
Machine Instruction Boundary
* Limited integration often multiple products
* Operating system calls security product for access checks
* Access checked once per user request
* Function integrated into the OS/400
* Machine instructions perform access checks
* Access may be checked multiple times in a single user request

Figure 7-1. Contrast of Traditional Systems to AS/400

The following common operations use several instructions that reference an object using a system pointer:

- Running a CL command
- Calling a program
- Opening a file
- Sending a message stored in a message file

The AS/400 performs an access check on each instruction that uses a system pointer resulting in multiple access checks for a single CL command. On first reference to an object, the operating system may store the user′s authority in the pointer, which eliminates the search for user′s access on subsequent references using the authorized pointer. For example, when a program opens a data base file, the user′s access is checked and OS/400 data management stores the user′s authority in the system pointer. Subsequent read and write operations to the open file do not require searching for authority. Open files are allocated so that authority to the file does not change. Most CL commands are not able to retain the authorized pointer after the command has completed. As a result, each time the command is issued the access checks are repeated.

Figure 7-2 on page 7-3 shows the number of access checks for an example CL program that might be used to update the invoice number. The program retrieves a data area, increments the number, and places the updated number back in the data area. The seven access checks that are made each time the program is called are shown to the right of the program. The objects checked are shown to the right of the program. The seven access checks that are made each time the program is called are shown to the right of the program.

```
       CL PROGRAM                               Checks user access to

PGM                                              1. SAMPLE    program
  DCL &value   *DEC  (15 5)
  RTVDTAARA (LIB1/TEST) VALUE(&value)            2. RTVDTAARA command
                                                 3. LIB1      library
                                                 4. TEST      data area
  CHGVAR   &value  (&value + 1)

  CHGDTAARA (LIB1/TEST) VALUE(&value)            5. CHGDTAARA command
                                                 6. LIB1      library
                                                 7. TEST      data area
ENDPGM
```

Figure 7-2. Access checks by CL Program

You cannot control the number of times that the AS/400 checks access, but you can select a security implementation that minimizes the overhead for each access check. An efficient security design is the focus of the second part of this chapter.

7.1.1 Object Structure

Figure 7-3 shows that AS/400 objects have two parts:

- An object header common to all objects
- A functional part

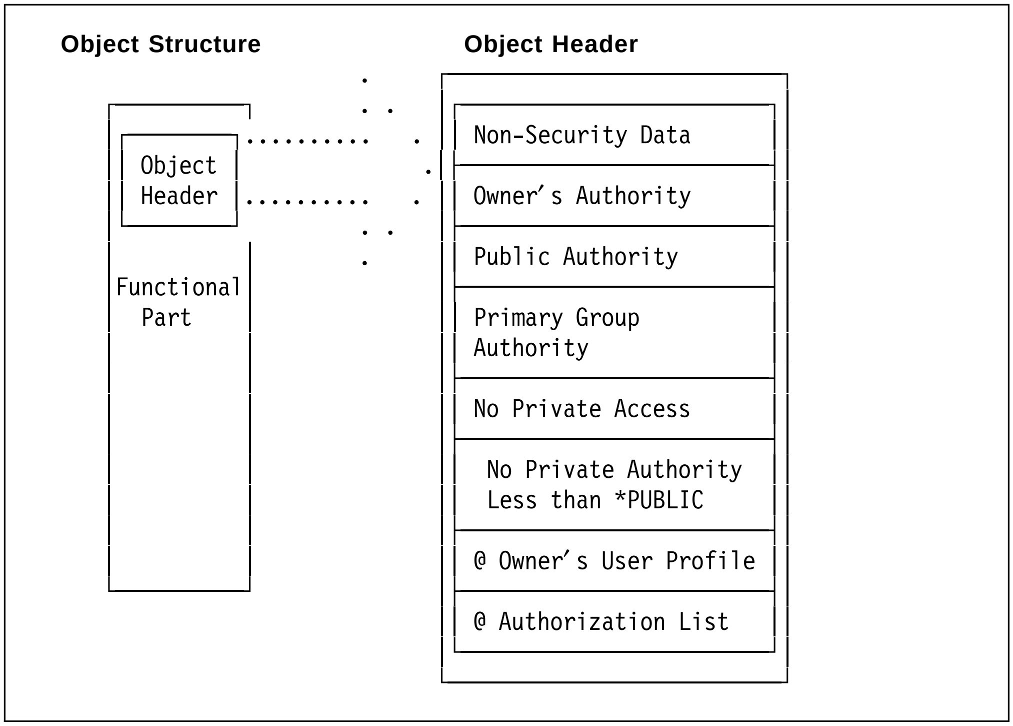


Figure 7-3. Object Structure and Object Header

The functional part of an object is different depending on the object type. For example, programs have instructions and files have records and so on. The object header is the same for all objects. The system access control functions use the common object header to validate access to all objects. The security related fields in the object header are:

Non-Security Data: The object header contains several fields that give the object type, name, and related information.

Public Access: This field stores a bit encoded representation of the *PUBLIC authority for the object.

Owner′s Access: This field stores a bit encoded representation of authority for the owner of the object.

Primary Group Access: This field stores a bit encoded representation of authority for the primary group (if any) of the object.

No private access: This flag indicates that there are no users with any access to the object. The *PUBLIC and owner′s are the only authorizations to the object.

No private authority less than *PUBLIC: This flag is set off if any user is given less authority than the *PUBLIC authority. Users may have private authorities that allow additional access and this flag will remain on. This flag is used to optimize the access control checks. **If this flag is on and the public access is adequate, the system does not need to search individual user profiles.**

Authorization list pointer: This pointer contains the address of the authorization list that secures the object. The pointer is null if there is no authorization list. When the object is secured by an authorization list, the access control search includes the authorizations to the authorization list.

Owner user profile pointer: This pointer contains the address of the individual or group profile that owns the object. This pointer is used to determine if the user of the current job is the owner of the object.

7.1.2 Private Authority

Figure 7-4 on page 7-5 shows that authority granted to the object owner and the *PUBLIC is stored in the object header. The owner′s and *PUBLIC authority is established when the object is created. The GRTOBJAUT (grant object authority) command can be used to grant access to other users. Access granted to other users is stored in the user profile of the user that is granted access.

The amount of CPU resource required to retrieve authority stored in the user profile is many times more than that required to access the object header. The security search process is more efficient when access can be determined from the public or owner′s authority. The second part of this chapter gives some rules that lead to an efficient security design.

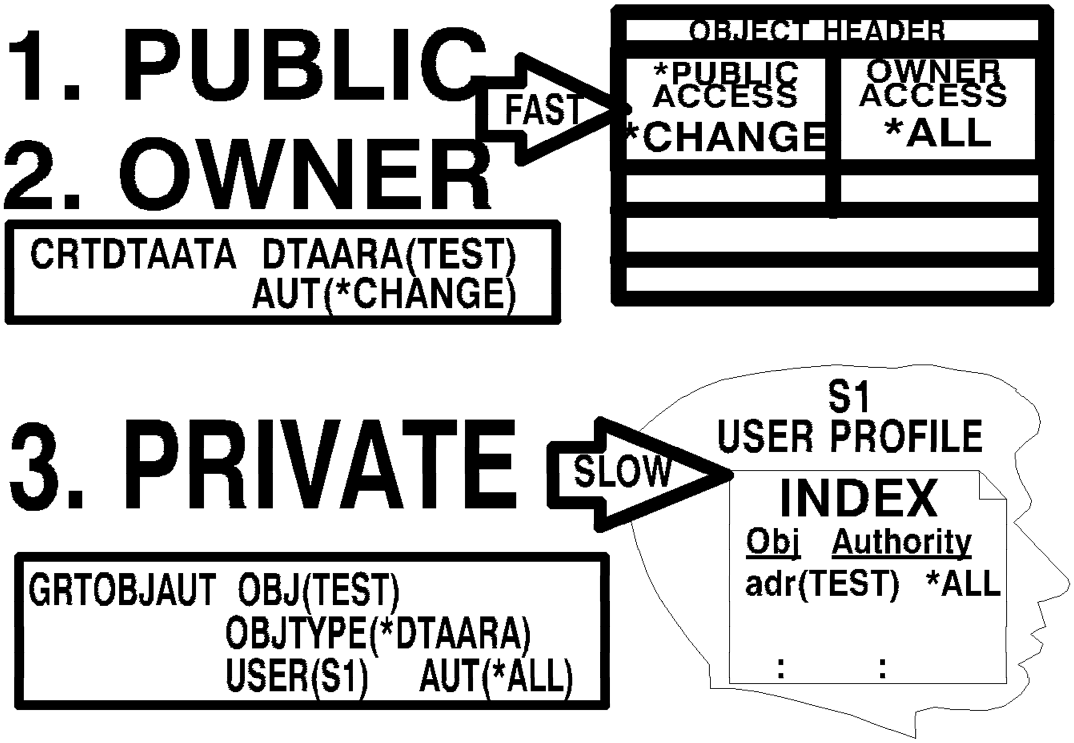


Figure 7-4. Three Forms of Authority

7.1.3 Authority Lookup

An *Authority lookup* operation is the term used to describe the search of a user profile for access to an object. A large number of authority lookup operations per second will have an impact on system performance. The tables on pages 7-6 and 7-8 give the approximate amount of CPU resource used for authority look up exceptions. These tables are reprinted from the IBM Performance Analysis Workshop notes.

You can determine the number of authority lookup operations for your system from the Performance Tools/400. If you do not have the performance tools product installed, you can determine the number of authority lookup operations using the OS/400 command STRPFRMON (start performance monitor). The STRPFRMON command starts a job that samples the system periodically and records the number of authority lookup exceptions during the period in the file QAPMSYS. Figure 7-5 on page 7-6 shows a report created from this data using the Query/400 product. The number of authority lookup exceptions per second is determined by dividing the number of lookup operations, SYAUTH, by the length of the interval, INTSEC. This number is used to find the percentage of CPU devoted to security lookup operations from the tables on 7-6.

During the interval ending at 19:46 the report shows there was an average of 95.16 lookup operations per second. The test was run on a AS/400 model F25, Table 7-4 on page 7-8 indicates that about 11% of the CPU resources were used for authorization lookup operations. An authority lookup overhead of 3% or less is a reasonable CPU overhead for security. Authority lookup impacts to CPU above 5% should be investigated. The rules the following section will assist you with your security design.

The authority lookup operations for your system will not vary as much as in the sample report on Figure 7-5. This report is based on actual data recorded when running a program similar to the program on Figure 7-2 on page 7-3. The program was modified to loop changing the retrieving and changing a data area 10000 times. The name of the library and data area were passed as parameters. Different settings of authority on the data area and library used to observe the authorization lookup changes. This was run on a dedicated system with a delay time of 10 minutes between each of the different tests. The different runs and authority settings are indicated on the report.

```
Time   Authority Authority CPU Time
       Lookup     Lookup    Used in
       per sec.    Count   Millisec

19:36    .00          0       513
19:41  21.96      6,565     1,522
19:46  95.16     28,452     4,728
19:51  16.67      4,985     1,267
19:56    .00          0       531
```

Figure 7-5. Sample Authority Lookup Report

Table 7-1. Number of Authority Lookup Operations

Authority of Object	Authority of Library		
	Library list *LIBL	**Named Library Public Only**	**Named Library Private authority**
Public Only	0	20,000	20,000
Private More	0	20,000	20,000
Private Less	20,000	40,000	40,000
AUTL More	20,000	40,000	40,000
AUTL Less	20,000	40,000	40,000
Private & AUTL	40,000	60,000	60,000

Public Only Object has only *PUBLIC access no private or authorization list

Private More *PUBLIC access is adequate but some users have private access that gives more than *PUBLIC

Private Less *PUBLIC access is adequate but some users have a private access that is less than *PUBLIC authority.

AUTL More Object is secured by authorization list that gives some users more access than the *PUBLIC authority.

AUTL Less Object is secured by authorization list that gives some users less access than the *PUBLIC authority.

Private and AUTL Object is secured by both private authority and an authorization list

Table 7-2. Models B and C CPU Overhead for Authority Lookup Operations

Lookups per second	System Model													
Auth	B10	B20	B30	B35	B40	B45	B50	B60	B70	C04	C06	C10	C20	C25
10	5	3	4	3	3	2	2	1	1	5	4	4	3	2
25	13	7	10	8	7	6	4	2	2	13	10	10	7	6
50	25	15	20	17	14	11	8	5	4	25	20	20	14	12
100	51	30	39	34	28	22	15	10	7	51	39	39	28	24
200		60	78	68	57	44	31	20	15		78	78	57	49
300		90			85	67	46	30	22				85	73
400						89	62	40	30					97
500							77	50	37					
600							93	60	45					
700								70	52					
800								80	60					
900								90	67					
1000								100	75					

Table 7-3. Model D CPU Overhead for Authority Lookup Operations

Lookups per second	System Model											
Auth	D02	D04	D06	D10	D20	D25	D35	D45	D50	D60	D70	D80
10	4	3	3	3	2	2	2	1	1	1	0	0
25	10	8	7	7	6	4	5	3	3	2	1	1
50	20	17	13	14	11	8	10	7	6	3	2	1
100	39	34	27	28	22	15	20	14	11	6	5	3
200	78	68	54	57	44	31	41	28	23	13	9	5
300			81	85	67	46	61	41	34	19	14	8
400					89	62	82	55	45	25	19	11
500						77		69	57	31	23	13
600						93		83	68	38	28	16
700								96	79	44	33	19
800									91	50	37	21
900										57	42	24
1000										63	47	27
2000											94	54
3000												81

Table 7-4. Models E and F CPU Overhead for Authority Lookup Operations

Lookups per sec	**System Model**													
Auth	E02	E04	E06	E10	E20	E25	E35	E45	E50	E60	E70	E80	E90	E95
10	3	3	2	2	2	1	2	1	1	1	0	0	0	0
25	8	7	5	5	4	3	4	3	2	1	1	1	0	0
50	17	13	10	10	8	6	8	5	4	3	2	1	1	1
100	34	27	20	20	15	13	15	11	8	5	4	2	2	1
200	68	54	41	39	31	25	31	22	16	11	8	4	3	3
300		81	61	59	46	38	46	33	25	16	11	6	5	4
400			82	78	62	51	62	43	33	21	15	9	6	5
500				98	77	64	77	54	41	26	19	11	8	6
600					93	76	93	65	49	32	23	13	9	8
700						89		76	58	37	26	15	11	9
800								87	66	42	30	17	13	10
900								98	74	47	34	19	14	11
1000									82	53	38	21	16	13
2000											76	43	31	25
3000												64	47	38
4000												86	63	51
5000													78	63
6000													94	76
7000														89
Auth	F02	F04	F06	F10	F20	F25	F35	F45	F50	F60	F70	F80	F90	F95
10	3	2	2	2	1	1	1	1	1	0	0	0	0	0
25	7	5	4	4	3	3	3	2	1	1	1	0	0	0
50	13	10	8	8	6	5	5	4	3	2	1	1	1	0
100	27	20	15	15	13	11	11	9	5	4	3	2	1	1
200	54	41	31	31	25	22	22	17	11	7	5	3	2	2
300	81	61	46	46	38	33	33	26	16	11	8	5	3	3
400		82	62	62	51	43	43	35	21	15	11	6	5	4
500			77	77	64	54	54	43	27	19	13	8	6	5
600			93	93	76	65	65	52	32	22	16	9	7	6
700					89	76	76	61	38	26	19	11	8	7
800						87	87	69	43	30	22	12	9	8
900						98	98	78	48	34	24	14	10	9
1000								86	54	37	27	15	12	10
2000										74	54	30	23	20
3000											81	46	35	30
4000												61	47	40
5000												76	58	50
6000												91	70	60
7000													82	70
8000													93	80
9000														90

7.1.4 Authority Search Order

The authority to objects can come from multiple sources including public authority, private authority, group profiles, authorization lists, *ALLOBJ special authority, and programs that adopt their owner′s authority. The system selects the first authority found in an ordered search starting with the individual user profile, next the group(s) profile and finally *PUBLIC. If the authority found is not adequate then any adopted authority is searched. The search order is:

USER PROFILE

1. Does user profile have *ALLOBJ special authority
2. Does user profile have explicit authority to the object
3. Does user profile appear on the authorization list of object

GROUP PROFILE (repeated for each group the user is a member of)

4. Does group profile have *ALLOBJ special authority
5. Does group profile have explicit authority to the object
6. Does group profile appear on the authorization list of object

PUBLIC

7. Use the object *PUBLIC authority unless *AUTL is indicated .
8. Use the *PUBLIC authority from the authorization list

PROGRAMS THAT ADOPT

9. Does program owner′s profile have *ALLOBJ special authority
10. Does program owner′s profile have explicit authority to the object
11. Does program owner′s profile appear on the authorization list of the object

The system uses the flags *No private authority* and *No authority less than public* to eliminate some of these steps. The *Security - Reference* publication gives the detailed search order and the impact on performance. This rather complex search order can be reduced to a few simple rules that you can use when securing your objects.

7.2 Security Design Options

This section discusses how you can secure objects to minimize the performance overhead associated with access control checks. The security design goal is both *efficient* and *effective* control of access to system resources. An efficient security option would be to give all users *ALLOBJ access or have the *PUBLIC authority of *ALL for every object. This is very efficient but would not be effective. The following are some rules that are both effective and efficient.

7.2.1 Public Authority

Rule 1

Only secure objects that require protection.

Authority to objects are similar to the door locks used to limit access to rooms. You do not put locks on every door in your home or office. If you had to lock and unlock a door every time you moved between rooms, it would be a slow process and a waste of time. In a similar manner, you should not secure every object in the system. When objects are restricted the system will check user′s access on each reference which can impact performance. Only the objects that need to be secured should have restricted access. When ever possible make *PUBLIC access adequate for most operations. The system uses the **No private access** flag in the object header to eliminate the need to check individual profiles for access.

Rule 2

Avoid giving users less authority than *PUBLIC access.

Giving any user less access than *PUBLIC will cause the system to perform an individual access check for every user. Giving a user less access than *PUBLIC is like placing a guard at a door to prevent one person from entering. The guard requires every person that enters the door to produce identification. The guard will allow access to all users but the restricted individuals. However, every user is required to produce identification each time they enter the door.

In a similar manner, if you give users less access than *PUBLIC for an object the system must check all users. The flag **No private authority less than *PUBLIC** eliminates the need to search individual user profiles when *PUBLIC authority is adequate for the operation. Access to *PUBLIC authority in the object header can be done with less CPU resource than the searching for private authority in user profiles. If a single user has less authority than *PUBLIC access the flag is turned off. This results in an authority search for every user that has access to the object.

It is best to reduce *PUBLIC access for an object to the lowest access rather than giving a few users less access than *PUBLIC. Some users may require more access than the *PUBLIC. The system will search the user profiles only when the operation requested requires more authority than *PUBLIC. For operations where *PUBLIC authority is adequate no search of the user profile is required.

The impact of the **No private authority less than *PUBLIC** setting is illustrated in Table 7-1 on page 7-6. There are no authority lookup operations when there are no private authorities, or all private authorities were higher than the public authority. When a user was given less authority than *PUBLIC, the authority count increased to 20,000. The 20,000 represents two authority lookup operations for the data area (once for the RTVDTAARA and once for the CHGDTAARA command) each time the program loops.

Rule 3

When granting users more access than *PUBLIC, use private authorities not authorization lists.

Users can be granted more access than public using explicit private authority or using of an authorization list. If an authorization list is used, the performance advantage of the **No private authority less than *PUBLIC** flag is lost. **An authorization list should not be used to give users more authority than *PUBLIC authority.** Users should be explicitly granted access to the object rather than using an authorization list. When an object has an authorization list there will be a check of the authorization list.

The difference in the number of authority lookup operations between a private authority and an authorization list is illustrated in Table 7-1 on page 7-6. When we used private authority to give users more access than public, there were no extra authorizations. When an authorization list was used, the authority count increased to 20,000.

7.2.2 Library Security

Rule 4

Group objects into libraries and restrict access to libraries.

Figure 7-6 shows the number of access checks categorized by the library and object. The number of times that the object is referenced represents 50% of the authority checks in the program. To access an object requires authority to both the library and the object in the library. Users must have at least *USE access to a library to access the objects within the library.

The security method called *library security* restricts access to the library, but not to the objects in the library. Library security has three advantages:

1. Restricts access to users authorized to the library,
2. Improves performance by limiting authority lookup operations to the library,
3. Simplifies security management by protecting multiple objects within a single library.

CL PROGRAM	Number of access checks		
PGM	Total	Library	Object
DCL &value *DEC (15 5)			
RTVDTAARA (LIB2/TEST) VALUE(&value)	2	1	1
CHGVAR &value (&value + 1)	0	0	0
CHGDTAARA (LIB2/TEST) VALUE(&value)	2	1	1
ENDPGM			

Figure 7-6. Access Checks by Library and Object

Rule 5

For performance reasons you may want to use *LIBL rather than the specific library name.

When a library is placed on the library list (*LIBL), the system checks the user′s access to the library. OS/400 stores the user′s authority in the library pointers on the library lists. Storing authority in the pointer eliminates the authority lookup operations when these *LIBL pointers are used.

If the program in Figure 7-6 is changed to use *LIBL rather than the specific library name LIB2, the access checks to the library are eliminated. Compare the number of authority lookup from the performance run on Table 7-1 on page 7-6. When an explicit library was named there is an additional 20,000 lookup operations. Each of the runs where an explicit library name was specified resulted in additional authorization lookup operations.

Rule 5 is easy to implement because most CL commands default to use *LIBL to access objects. This default has the performance advantage of only checking users access once when the library is added to the library list. Use of the *LIBL and library security (Rule 4) can dramatically reduce the number of authority lookup operations.

Note: This can be a security exposure when calling programs from programs that adopt, or if the library list is not secured from being changed.

7.2.3 Object Ownership

Rule 6

Objects in the production environment should be owned by user profiles, not by group profiles.

Rule 7

The same user profile should own both the programs that adopt and the objects referenced by the program.

The authority check performance for objects referenced by programs that adopt is improved when the objects referenced by the program are owned by the same user profile that owns the adopting programs. This allows the system to take advantage of the object owner′s authority stored in the object header.

Rule 8

Use authorization lists or private authority to secure objects. Do not use both methods.

Table 7-5 on page 7-13 shows the maximum authority lookup operations when a user is not authorized to the object. When the user is authorized, the number of lookup operations may be less depending upon where the system locates the needed access. You will note that row 8 has a potential of four lookup operations which is double either row 7 or row 6.

Row	Private authority	Authorization List	Group profile	Number of lookups	Order of lookup [2]
1	NO	NO	N/A	0	
2	yes [1]	NO	N/A	0	
3	YES	NO	NO	1	1
4	NO	YES	NO	1	2
5	YES	YES	NO	2	1 2
6	YES	NO	YES	2	1 3
7	NO	YES	YES	2	2 4
8	YES	YES	YES	4	1 2 3 4
Note: [1] All users have more authority than the *PUBLIC authority and the *PUBLIC authority is adequate for the operation.					
Note: [2] Authority can come from any of the following: 1. Object private access from the user profile 2. Authorization list access from the user profile 3. Object private access from the group profile 4. Authorization list access from the group profile					

Table 7-5. Authority Lookup in Different Cases

7.3 Conclusion

This chapter has focused on how security impacts performance. System performance can be affected by many other factors and often is a combination of several factors. If your system is experiencing poor performance and the authority exceptions are low, then there may be other factors. IBM offers an AS/400 Performance Analysis Workshop that discusses methods to identify performance problems and understand the multiple components of system performance.

Chapter 8. C2 Overview and Considerations

This chapter gives an overview of what is required to implement C2 security on your system. You must read *Guide to Enabling C2 Security* if you are considering a C2 implementation.

8.1 What Is C2 Security Certification?

The United States Department of Defense (DoD) has defined standards for computer operating system security. These are defined in the *Trusted Computing Systems Evaluation Criteria (TCSEC)*, which is also known as the ″Orange Book″.

The AS/400 at V2R3 is designed to meet the C2 criteria as defined by the TCSEC. The AS/400 is currently in the midst of evaluation. The C2 level is one of the worlds most accepted and rigorous for computer security in non-military (mandatory) environments. To complete the government evaluation, the AS/400 is going through extensive analysis and testing by an independent team.

An objective of the orange book is to provide a set of off-the-shelf computers that meet a defined level of security and integrity that government installations can use as building blocks to build secure systems. The AS/400 has defined a configuration that is both evaluatable and that meets the C2 criteria. This configuration, described in detail in the *Guide to Enabling C2*, is the base building block for users to begin building their own unique secure system.

All IBM Licensed Programs that run in privileged (system) state and are not part of the evaluated configuration, plus applications that adopt additional authority pose a potential security/integrity risk. These LPs and applications are restricted from a C2 system since they have not been evaluated. Users who need or want an LP or application that runs in privileged state should do their own evaluation or make a risk assessment. In addition, the AS/400 has been evaluated with certain objects being restricted and specific values for certain system values. Again, users should change these from a known secure state, only after doing a risk assessment. Users should take advantage of the QALWUSRDMN system value to allow trusted applications to share data while restricting untrusted users.

Users who need a system that meets C2 should follow the Guide in detail. Those who want a secure system should consider configuring to C2, then making changes needed to meet their business needs and/or cost constraints. This will provide a secure known base with an evaluation of risks associated with the deviations.

Communications is not a part of the C2 definition in the Orange book. A separate network interpretation (TNI) is available for given networks and is summarized later in Appendix J, “Trusted Network Interpretation (TNI)” on page J-1.

8.1.1 Changes to OS/400 for C2

The enhancements to OS/400 security to support C2 are as follows:

- QSECURITY level 50
- New system values: QAUDFRCLVL, QAUDENDACN, QALWUSRDMN
- User profile special authority: *AUDIT
- The ability to audit:
    - System-wide security actions
    - The access of an individual object
    - The actions of an individual user

8.1.2 What Are the TCSEC Security Levels?

The TCSEC defines six levels of security as follows:

Division	Classes
D	Minimal Protection
C	1 Discretionary Security
C	2 Controlled System Access
B	1 Labeled Security Protection
B	2 Secured Protection
B	3 Security Domains
A	1 Design Verified and Certified

See J.2, “Interpretation of TCSEC” on page J-3 for a more detailed description of protection offered by these security levels.

8.2 Implementation of C2 on a System

The *Guide to Enabling C2 Security* describes the formal procedures and steps to be taken for implementation. The following tables will give you an overview of the specific actions required to implement C2 on your system. For a system to meet C2 means compliance to rules in two major categories:

- A specified control *must* be implemented.
- A specified control *is able* to be implemented, but doesn′t *have* to be implemented at a particular point in time.

You should, however, implement the major part of these actions, even if you do not plan to implement C2. They represent a sound security policy.

The following tables reflect actions that have to be performed based on these two requirements.

What Has to Be Done	Why This is Important for C2
QSECURITY level 50	Performs additional parameter validation during program calls.
QALWUSRDMN set to QTEMP	So that *USRxxx-type objects cannot be used to pass data between users. QTEMP must be the only library in QALWUSRDMN.
*PUBLIC authority set to *EXCLUDE on libraries containing application objects, or on individual objects within libraries	Only users who are explicitly authorized should be able to access application objects.

What Has to Be Done	Why This is Important for C2
CRTAUT on application libraries set to *EXCLUDE	Users must be specifically authorized to new objects created in application libraries.
For user profiles that are group members, set OWNER to *USRPRF and GRPAUT to *NONE	Gives strict control of ownership. There is little or no control when a group profile is owner.
Delete all authority holders from the system	Authority holders store the authorizations to files, even when the files are deleted from the system. They give the authorizations back when the files are restored/recreated. They do not meet the requirement for explicit authority within C2.
Create functional user profiles for users who require special authorities such as *SAVSYS	A user profile must only have the authorizations needed to do the job, no more. Special authorities must only be used for special functions by trusted system users, and only when doing these special functions.
Only base OS/400, SQL/400, Query/400, AFP Fonts, and IBM Common Cryptographic Architecture Services/400 are permitted on the system	They run in system state and have been evaluated as part of the trusted computing base.
Only SEU (not the rest of the Application Development Tools), permitted on the system	It runs in system state and has been evaluated for C2 security.
Save all objects off your system, and perform a scratch install of the base operating system only with the IBM tapes. Security level 50 must be implemented immediately after the OS/400 install	Wipes out all the existing data. IBM tapes have not been tampered with.
Change the default passwords for IBM-supplied user profiles	Must only be known to those who need to know.
Change the default passwords for the DST user profile	Must only be known to those who need to know.
Install only C2-certified PTFs	The PTFs on tapes with feature code 1920 have been evaluated for C2 security.
`CHGCMDDFT CMD(CRTLIB) NEWDFT('AUT(*EXCLUDE)')`	New libraries will get created such that only authorized users are allowed to use these libraries.
Restrict physical access to devices and media used for save and restore operations	Only trusted system users should be able to take a backup copy of data or programs, or restore programs to the system.
Set *PUBLIC *EXCLUDE on all SAVxxx and RSTxxx commands	Only trusted system users are authorized to perform these operations.
Set *PUBLIC *EXCLUDE on CRTLINxxx commands	C2 does not allow communication lines, but if you make a risk assessment, only trusted system users should be able to create communication lines.
Set *PUBLIC *EXCLUDE on WRKCFGSTS and VRYCFG commands	Only trusted system users should be allowed to control the status of communication lines.
Set *PUBLIC *EXCLUDE on the CRTAUTHLR command and ensure that no users have explicit authorities to this command	C2 does not allow authority holders. This will prevent them from being created.
Cannot be system unit model B, C, and D10 and D20	These models have no Hardware Storage Protect feature.
System cannot have attached PCs	C2 does not allow the use of PCs.

What Has to Be Done	Why This is Important for C2
No configured communications lines	C2 does not allow communication lines.
No workstation-attached printers that allow direct screen print	All print data must be handled by OS/400 spool control.
When using DLTUSRPRF, spool files must be deleted	C2 requires control of ownership including spool files.
Perform periodic audits of audit journal	To ensure that any deviations from the points above are monitored.

Table 8-1. Settings Required on a C2 System

What Has to Be Able to Be Done	Why This is Important for C2
Setup Audit journal and receiver	C2 requires auditing capabilities.
Set QAUDLVL sysval to *SAVRST, *SECURITY, *PGMFAIL, *AUTFAIL	Records security-related events in the audit journal.
Set QAUDCTL sysval to *AUDLVL	Starts journaling to the audit journal.
Set up object auditing for tape and diskette device descriptions	Only trusted system users may use tape and diskette.
QAUDENDACN set to *PWRDWNSYS	The system cannot operate if security-related events are not logged.
QAUDFRCLVL set to 1	Force security-related events to the audit journal immediately.

Table 8-2. Settings that Must Be Able to Be Implemented

Typically, most sites would not go to the extreme situation to implement all of the points described above. The *Guide To Enabling C2* gives some more appropriate values for the system values and other settings described.

8.3 Performance Impacts of C2 and Level 50

As described in Chapter 2, “System Values and Network Attributes” on page 2-1, the implementation of security level 50 causes additional parameter validation to occur. There could be a significant performance impact on interactive and batch users of the system in this environment.

8.3.1 Impact on Daily Operations

When at security level 50 the QTEMP library is handled differently from level 40 and lower. At level 50, when a job terminates abnormally, the QTEMP library is not ″cleaned up″ as it is on level 40 and lower.

The result is that the RCLSTG command must be run more frequently to reclaim the disk space and the addresses. Before RCLSTG command is run the QRCL library must be added to QALWUSRDMN, and it must be removed afterwards.

8.4 Recommendations

C2 security is for AS/400 sites who wish to achieve extremely high standards of security. If you must do something that deviates from C2, first do a risk assessment.

To achieve better control over *any* AS/400 system, you should attempt to implement most of the steps for C2.

Chapter 9. AS/400 Cryptography

In this chapter we will look at the way cryptography is implemented on the AS/400, including:

- An overview of the various components of cryptography
- Planning and documentation
- Setting up an AS/400 for cryptography, and
- Key management

This chapter will focus primarily on hardware cryptography features used with the IBM Common Cryptographic Architecture Services/400.

Manuals

Information about cryptography is documented in many different manuals. The *Common Cryptographic Architecture Services/400 Installation and Operator's Guide* is mandatory reading. Before you start to plan to use cryptography on the AS/400 you should order the manuals about Transaction Security Services referenced in "Related Publications" on page xxvi.

9.1 Introduction to Cryptographic Support

Data security threats can be minimized by implementing sound computer security practices. The AS/400 provides excellent security built into OS/400 with access control and audit capabilities. However, once the data is transmitted over a network or copied to an offline medium, the access control facilities of the host computer can no longer protect it. Once the data is outside the boundaries of the processor, cryptography is the only means of protection. Fortunately, properly implemented cryptography is very effective, but it requires skill, planning, documentation, and proper management of the cryptographic keys used.

9.2 IBM Common Cryptographic Architecture Services/400 (CCAS/400)

AS/400 Hardware Cryptography is part of a family of IBM Common Cryptographic Architecture comprising of products known as the IBM Transaction Security Services, or TSS. In fact, the AS/400 uses the same hardware components as PCs running DOS or OS/2, the RS/6000, and the channel attached 4753 Network Security Processor control unit.

It follows a published architecture which has been defined by leaders in the field of cryptography. Since products on other IBM platforms also conform to this architecture, the AS/400 solution is compatible with these IBM platforms.

The AS/400 provides the same programming interface as the other TSS family members. The cryptographic solutions across these platforms fully inter-operate. Applications are highly portable between platforms.

Generally, cryptography can be implemented at three different levels:

Link level: All data is enciphered and deciphered at the level of the physical connection between nodes in a network. At each system that the data flows through, the data must be deciphered and enciphered.

IBM Common Cryptographic Architecture/400 does not support Link Level Encryption.

Session level: Session Level Encryption is a Systems Network Architecture (SNA) protocol that provides a method for establishing a session with a unique key for that session. This protocol establishes a cryptographic key and rules for deciphering and enciphering information in a session.

IBM Common Cryptographic Architecture/400 does not support SLE.

Field level: The user-written applications use APIs to request cryptographic services. CCAS/400 only has support for field level enciphering.

9.3 Description of Cryptographic Functions

The following cryptographic functions are common on most platforms.

The Data Encryption Algorithm (DEA): DEA is defined in the American National Standards Institute (ANSI) X3.92-1981 standard. It is often called the Data Encryption Standard (DES). The DES defines a symmetric algorithm. The algorithm uses the same key for enciphering plaintext to ciphertext and for deciphering ciphertext to recover the original plaintext.

Export Restrictions for DES: There is a restriction on the distribution of DES products. The U.S. government allows the IBM DES products to be sold freely only for U.S. and Canadian customers, and generally for financial institutions worldwide and for international subsidiaries of U.S. corporations. Other customers must get a permit to use DES products. This permit must be requested from the U.S. government. For information regarding a specific country, the IBM export coordinator in the country must be consulted.

Encryption and Decryption: Data that is to be kept confidential can be enciphered. The enciphered form of the data (ciphertext) can later be deciphered to recover the original data (plaintext). The ciphertext can be stored on disk or an external medium, or sent to another system. An adversary cannot recognize or understand the data without access to the key used to decipher. The Data Encryption Algorithm (DEA) mentioned above is used to perform the encryption and decryption. DEA resides and operates in the cryptographic adapter.

Encryption and decryption are basic functions of IBM Common Cryptographic Architecture Services/400.

Cryptographic Keys: The DES algorithm operates on 64 bits at a time (8 bytes of 8-bit-per-byte data). The key has 56 bits and it is conventionally presented in 8 bytes. Each byte of the key contains 7 bits of key information plus a parity bit (the low-order bit in the byte). By convention, the parity bit is set so that there are an odd number of one-bits for each key byte. The parity bits do not participate in the DES algorithm. For example, a valid key could be:

```
6D61 CE32 C201 6237
```

Table 9-1 on page 9-3 shows the bit pattern of a few characters.

Table 9-1. Example of Bit Patterns in Characters

Character	Hex. value	Bit pattern	No of 1-bits	Key character
	6D	0110 1101	5	Yes
	61	0110 0001	3	Yes
	CE	1100 1110	5	Yes
	32	0011 0010	3	Yes
A	C1	1100 0001	3	Yes
B	C2	1100 0010	3	Yes
C	C3	1100 0011	4	No
1	F1	1111 0001	5	Yes
2	F2	1111 0010	5	Yes
3	F3	1111 0011	6	No

The secrecy and reliability of DES-based cryptography is strongly related to the secrecy, control, and use of DES-keys. Thus key management is critical.

Key Management: Cryptographic keys (DES or PKA keys) are considered the most important resource within the security system. Good key management must provide protection of the cryptographic keys, and is essential when implementing cryptography. Management of the keys requires careful planning and implementation, and every step must be documented.

With regard to key protection, the IBM Common Cryptographic Architecture Services/400 product has been designed to prevent keys from being available, both physically and logically, in their clear form except when they are placed in one of the protected registers or buffers of a cryptographic adapter, or during key generation. Hardware tamper-resistant features are reviewed in the "Cryptographic Adapter" on page 9-7.

Automated key storage and retrieval within the internal cryptographic adapter is provided by the IBM Transaction Security Services. The methods for distribution of keys across a network can vary significantly from institution to institution. The IBM Transaction Security Services products have been designed with flexibility to enable support for existing standards where applicable, in conjunction with user written key management programs.

The following operational and administrative functions should be well documented in an organization′s security plan:

1. Generation of keys
2. Key distribution, electronically and/or manually
3. Key usage control
4. Key installation
5. Key verification within the installation
6. Secure key storage
7. Key destruction
8. Periodic replacement of keys
9. Archival of keys

10. Monitoring the key generation, distribution, installation, archival, and destruction processes
11. Journaling and auditing of the security system
12. Journaling and auditing management
13. Reactions to security violations

The IBM Common Cryptographic Architecture Services/400 can directly support an organization′s efforts within items 1 through 7. The remaining aspects of key management must be addressed within the organization′s management policies, procedures, and application software facilities.

Message Authentication Code (MAC): MAC is a way to calculate a checksum over a given set of data. The data can be any type: numeric, alphabetic, packed, or binary. The purpose of the MAC is to be sure that the given set of data has not changed after it was produced. For example in the case of Automatic Teller Machine (ATM) transactions, it is more important that we know the message has not been altered than it is to keep the entire message confidential. This can be done using a MAC. If this same MAC can be generated from a set of data at another location or at another time, you can verify that the data in the received message is the same as the original. The MAC standard is based on DES and is calculated using a cryptographic key.

The IBM Common Cryptographic Architecture Services/400 offers MAC functions.

Modification Detection Code (MDC): MDC is an IBM published algorithm based on DES that uses a predetermined 112 bit key. This key does not have to be secret. The MDC calculation is basically the same as calculating a MAC. The content of the data together with a cryptographic key is used to calculate the MDC. MDC offers more protection than MAC, because MDC uses a longer key.

The IBM Common Cryptographic Architecture Services/400 offers MDC functions.

Managing Personal Identification Numbers (PIN): PINs are used by ATMs to identify a customer, but can also be used by other types of equipment. A PIN is usually four digits, for example, 0042. The PIN block is a 64-bit data structure that helps to protect a clear PIN from disclosure. Before a clear PIN is transmitted or stored, the PIN is normally formatted into a PIN block, which is encrypted. The PIN block contains the clear PIN and, depending on the format, one or more non-PIN digits. For description of each PIN-block format, see *IBM Transaction Security Services Programming Reference: Volume 1, Access Controls and DES Cryptography.*

The IBM Common Cryptographic Architecture Services/400 offers PIN management functions; it generates and verifies PINs, translates PIN block formats, and creates PIN blocks.

Public Key Algorithm (PKA): PKA is used to transmit encrypted DES keys between systems, and to generate and verify digital signatures. PKA uses two different keys, public and private. The public key is not secret, and is used for encryption. The corresponding key, called the private or secret key, is used for decryption. One widely known public key algorithm is called RSA, after the people who defined it - R. Rivest, A. Shamir, and L. Adleman.

The IBM Common Cryptographic Architecture Services/400 supports PKA functions.

No Export Restrictions for PKA: The PKA (Public Key Algorithm) is only used to encipher and decipher DES key-encryption keys, and to generate and verify digital signatures. PKA uses the RSA algorithm, so there are no export restrictions on this algorithm.

Signature Verification: When you sign your name, the signature verification feature compares the movements of the signature verification pen with reference data that is stored in your IBM Personal Security Card (PSC or smart card). The verification process tests the way in which the pen is moved rather than the appearance of the signature. Signature verification is considered to be stronger than PIN verification.

IBM Common Cryptographic Architecture Services/400 does not support signature verification on the 4754 SIU that is locally attached to the AS/400. However, signature verification is supported on TSS workstations.

9.4 IBM Common Cryptographic Architecture Services/400 Components

IBM Common Cryptographic Architecture Services/400 consists of a cryptographic processor, a cryptographic adapter, and controlling software provided by PRPQ 5799-FBR or 5799-XBY. It is available on all AS/400 D and later models, except for the D02 and E02. There is a maximum of one cryptographic processor per system.

9.4.1 PRPQ 5799-FBR or 5799-XBY

The PRPQ 5799-FBR is used on V2R3 systems, and 5799-XBY on V3R1 systems. These PRPQs provide the application programming interface, often called Security API or SAPI. The services of the cryptographic processor are controlled by user-written applications using these APIs. The PRPQ also provides for keys to be stored in the files QAC4KEYST and QAC4PKEYST in library QUSRSYS. All the keys stored there are in encrypted form. These keys are encrypted using a key called Master Key. The master key cannot be stored in encrypted form, because then another key would be needed to encrypt it. The master key is stored in unencrypted, or clear form, in the shielded module of the cryptographic adapter. Special measures are taken to protect it, and the customer must keep it secret.

If you order one of the PRPQs, you only get the software and the *Common Cryptographic Architecture Services/400 Installation and Operator's Guide*. The PRPQs contain:

- The QTSS library
- Four CL commands, which are restored into the QSYS and QTSS libraries
- QAC4KEYST file in QUSRSYS (for the DES Keys)
- QAC4PKEYST file in QUSRSYS (for the Public Key Algorithm keys (PKA))

9.4.2 The Cryptographic Processor

The cryptographic processor contains the PS/2 4755 Micro-Channel* Cryptographic Adapter inside an AS/400 book card.

Two different cryptographic processors are available:

- Feature #2620 is a full function Micro-Channel version. The adapter is part number 4755-014. U.S. export restrictions apply to this feature. See “Export Restrictions for DES” on page 9-2 for more information.
- Feature #2628 is a “commercial” function Micro-Channel version. The adapter is part number 4755-L14.

  This feature provides the same functions as feature #2620 except that the data confidentiality functions of encryption and decryption are provided using the new Commercial Data Masking Facility (CDMF) instead of the DES algorithm. CDMF is a new scrambling technique for data confidentiality and is intended as a substitute for the Data Encryption Standard (DES). CDMF complies with U.S. export regulations and will normally be available to most customers in most countries.

  CDMF uses a 64 bit key, but provides the effective strength of a 40-bit DES key. Feature #2628 is fully compatible with feature #2620 with the exception of the encryption and decryption commands. Interoperability on masked data between an AS/400 with feature #2628 and an AS/400 with feature #2620 or other full DES systems is not supported.

All cryptographic operations are performed in the cryptographic processor by a micro-processor in a special module shielded against attacks by an adversary. The micro-processor has specialized hardware to assist in high-speed DES calculations.

The master key is stored in clear form inside the shielded module. Any attempt to tamper with the module or to remove the cryptographic processor from the system results in the master key being overwritten with zeros. Because service on the cryptographic processor will require its removal from the system, the master key will be overwritten.

Important!

The security administrator must always have a backup copy of the master key with its version number. Loss of the master key, without a backup, will result in the loss of all encrypted keys and data.

All keys, except the master key, are stored in encrypted form in physical files. The files QAC4KEYST and QAC4PKEYST in library QUSRSYS are predetermined to be used for this purpose, but other physical files may be used. The keys are converted to clear form in order to be used in cryptographic operations. These operations occur only in the shielded module, and therefore keys never appear in clear form outside the protection of the shielded module. This provides for a very high degree of security for the encrypted keys and the data encrypted under them.

Another benefit of the cryptographic processor is that it handles the compute intensive cryptographic operations instead of the CPU. This can improve overall system performance when using cryptography.

Cryptographic Adapter

The IBM Cryptographic Adapter is a high-performance cryptographic component that supports a comprehensive set of cryptographic functions. There are different models of the cryptographic adapter depending on the:

- Full function or “commercial” function
- Micro-Channel architecture or PC-Bus version (when used in a PS/2 or a PC)

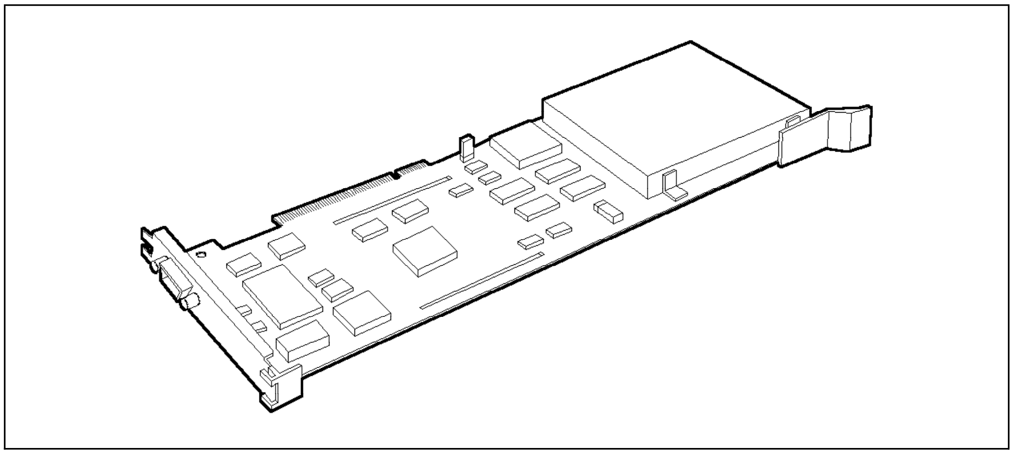


Figure 9-1. Micro-Channel 4755 Cryptographic Adapter (Model 4755-014)

The IBM Cryptographic Adapter contains an 80186 microprocessor, ROM, RAM and DEA (Data Encryption Algorithm) module which are protected from physical intrusion in a tamper-resistant package. It is also protected against electromagnetic and chemical attacks. If someone breaks the tamper-resistant package, the sensitive information in it, such as the master key, will be wiped out.

Using APIs in user written programs, a session can be set up between the cryptographic adapter in the AS/400 and the PSCs through the 4754 Security Interface Unit (SIU).

If the cryptographic adapter is removed from the AS/400 it will lose all its information. Thus, this should only be done when the cryptographic adapter or the cryptographic processor must be replaced or serviced. In V2R3 a backup of the master keys and the Master Key Version Numbers (MKVNs) must always exist. In V3R1 only a backup for the master keys must exist. In V3R1 the MKVNs are stored in QAC4KEYST and QAC4PKEYST files in QUSRSYS.

9.4.3 Other Optional Products

The supporting products listed below are optional. They are used together and must be ordered separately:

4754 IBM Security Interface Unit

The IBM Security Interface Unit has two models, the full-function version and the “commercial”-function version. The “commercial” function unit may be used in countries that cannot use the full-function version.

The IBM Security Interface Unit′s main functions are to configure the cryptographic profiles and the IBM Personal Security Cards (PSCs), and to load cryptographic profiles and keys from the PSCs into the cryptographic adapter in the AS/400.

Full function (4754-001) and “commercial” function (4754-L01) models of the 4754 are available. The full function model has the U.S. export restrictions.

The SIU is connected to a PS/2 (or PC) when the PSCs are configured. See Figure 9-5 on page 9-13. Afterwards this cable is removed and another cable connects the SIU to the cryptographic adapter to transport the data in the PSCs to the cryptographic adapter in the AS/400. See Figure 9-6 on page 9-15.

The IBM Security Interface Unit supports a comprehensive set of cryptographic functions (but only a subset of the functions in the cryptographic processor in the AS/400). It can process on its own or in conjunction with the cryptographic processor in an AS/400.

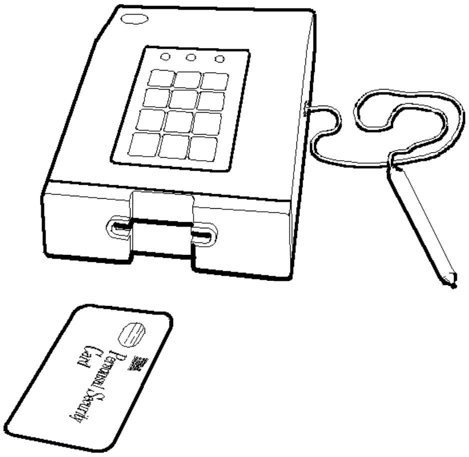


Figure 9-2. Security Interface Unit, Signature Verification Pen, and Personal Security Card

When used to configure PSCs the SIU is connected to the PC or PS/2 using the RS232 serial I/O port. When connected to the cryptographic adapter in an AS/400 a 9-PIN cable is used.

The IBM Security Interface Unit includes:

- Cryptographic processor
- Lights and audible tone
- Clock which can be set securely
- Battery for clock and for RAM contents

The cryptographic processor in the SIU provides main processing power for the SIU, controls the peripheral functions of the SIU, and controls the direct communications to the IBM Personal Security Card. It also provides 32KB of EEPROM (Electrically Erasable Programmable Read-Only Memory) and provides data buffering and secure information storage in the 8KB static RAM. The 8KB RAM has a battery backup. The RAM will be erased if tampering is detected by the tamper-resistant circuits.

The power switch for the IBM Security Interface Unit is at the back of the unit together with a voltage selection switch. Care must be exercised to set the voltage selection switch to the proper setting before turning the power on. The voltage switch should be covered with plastic tape after setting it correctly to prevent accidentally switching to the wrong voltage setting when powering the unit on or off.

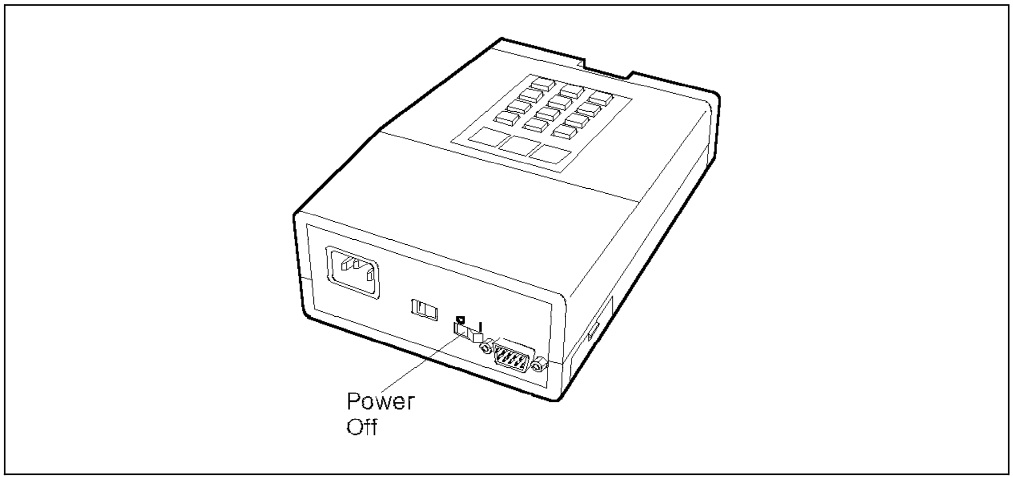


Figure 9-3. Security Interface Unit Power Switch

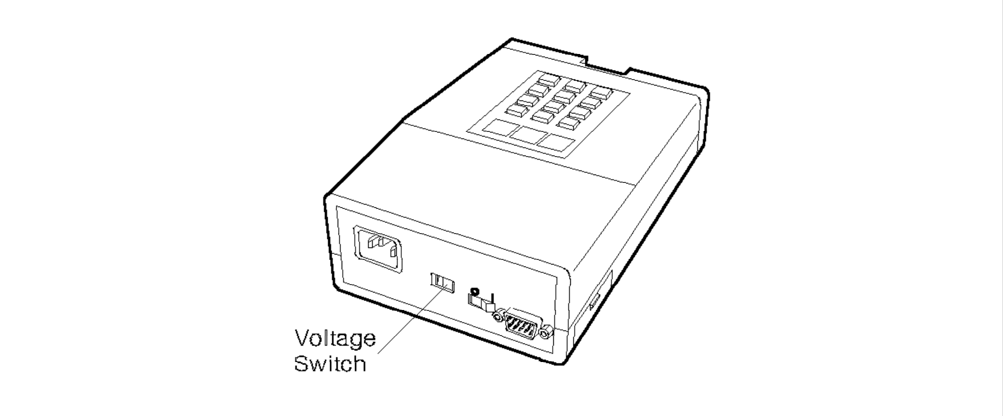


Figure 9-4. Security Interface Unit Voltage Selection Switch

IBM Personal Security Card (PSC, Smart Card)

The IBM Personal Security Card may be used to store cryptographic profiles, and to transport the master key and other cryptographic keys. Three IBM Personal Security Cards are shipped with each IBM Security Interface Unit. Additional IBM Personal Security Cards can be obtained by ordering one of the following part numbers:

- 41F9970 is the full function version, which has U.S. export restrictions.
- 82F5486 is the “commercial” function version; it does not have any export restrictions.

The IBM Personal Security Card is an integrated-circuit chip card that contains a single-chip security processor and storage facilities. The advantage of this single-chip design is that there are no exposed interfaces, as found in multiple chip environments, which might be tapped by an adversary. The Transaction Security System uses the IBM Personal Security Card to identify, authenticate, and authorize users, and to configure the cryptographic services to be used in an AS/400. It can also perform DEA-based cryptographic functions and serve as a portable file, for example to transport keys. A PSC can contain more than 4000 bytes of data in data blocks. Additionally, a magnetic stripe can be affixed to the IBM Personal Security Card if you wish to use the card in conjunction with other existing devices.

The IBM Personal Security Card communicates with the AS/400 through the IBM Security Interface Unit. The IBM Personal Security Card conforms to the evolving ISO (International Standards Organization) standards for the physical characteristics of integrated-circuit cards.

IBM PS/2

A PS/2 (or PC) is required to use the SIU and the PSCs. The PS/2 must have a serial (RS-232) port to attach to an IBM Security Interface Unit.

The Workstation Security Services Program (WSSP) must be installed on the PS/2. It can be obtained by specifying it as an option when ordering either the IBM Security Interface Unit or the IBM Cryptographic Adapter, or it can be ordered separately. The WSSP Hardware Initialization and Key Management Utility (HIKM) is used to initialize and configure IBM Personal Security Cards and to generate cryptographic profiles and master keys.

Workstation Security Services Program (WSSP): The Work Station Security Services Program runs in a PC or in a PS/2 with IBM PC DOS Version 3.3 or higher or an equivalent product. It also runs under OS/2 Versions 1.x and 2.x. See the *IBM Work Station Security Services Installation and Operating Guide* for more information about the operating system requirements.

The Work Station Security Services Program provides the following:

- Security API routines
- Security server
- Drivers
- Loader
- Sample routines and configuration information
- Work Station Security Services utilities
- Diagnostics

Work Station Security Services Program Utilities: The Work Station Security Services Program utilities reside and operate within a workstation. They are used to initialize and configure the Transaction Security System to the required needs.

The Work Station Security Services utilities manage the initialization, configuration data and profiles for all of the Transaction Security System components:

- Network Security Processor (NSP)
- IBM Cryptographic Adapter
- IBM Security Interface Unit
- IBM Personal Security Card

The Work Station Security Services utilities provide cryptographic key management, access to and backup of the TSS components, and enters signature references for the Personal Security card to enroll or verify users.

The Work Station Security Services utilities are composed of the following programs:

- Hardware Initialization and Key Management utility
- Batch Initialization utility
- Software Configuration utility
- Network Security Processor (NSP) support utility

9.5 Planning and Documentation

The very first thing you must do is to plan and document how you want to utilize the cryptographic services available. Start out by reading *Common Cryptographic Architecture Service/400 Installation and Operator's Guide*. This is mandatory. In your planning and implementation you will use other manuals too. Refer to "Related Publications" on page xxvi.

To use CCAS/400 you must:

1. Set up the PS/2 to configure the PSCs as shown in Figure 9-5 on page 9-13.
2. Connect the SIU to the cryptographic adapter in the AS/400 and transfer the data in the PSCs to the AS/400 as shown in Figure 9-6 on page 9-15 using the INZCS command.
3. Activate the cryptographic support by running the STRCS command.
4. Run the programs you have written to create the cryptographic keys your applications will require.
5. Use the cryptographic keys in your programs to encipher and decipher the data of your choice.

Planning

Here are some of the points you must consider:

- What do you plan to encrypt, and why, such as:
  - Messages
  - Fields in records in files on the disk
  - Backup
  - Data transferred to another computer, using communication lines or offline media
- Which cryptographic functions will be used.

  When you know how you will use the cryptographic support you must find out which cryptographic verbs and APIs that will be used. The manual *Transaction Security System Programming Reference: Vol I Access Controls and DES Cryptography* gives an overview over the cryptographic verbs that must be used, and the corresponding APIs; and *IBM Transaction Security Services Programmers' Reference: Volume II Public Key Cryptography for PKA Verbs* can be used to understand PKA functions.
- Defining the cryptographic profile.

  A cryptographic profile is a set of limitations that you must imply on the cryptographic services available. The cryptographic profile is generated in the PSCs and transferred to the cryptographic adapter in the AS/400 when the INZCS command is run. For more information, refer to *Workstation Security Services Installation and Operator′s Guide*.

- Which cryptographic keys and KEKs (Key-encryption keys) you plan to use.
- How will the keys be managed. This is extremely important.

  If you plan to encrypt fields in files on the disk, or on your backup media, your encrypted data will be lost forever if you do not have strict control of your keys. Note that:

  – Encrypted backup will have an impact on your restore strategy and your ″Plan to protect the business processes″.

  – When keys are changed you must know what to do if you must restore an object that was encrypted under another key than the present one.

  – If you are part of a network the other systems must be taken into consideration. If they too will use cryptography, cryptographic keys must be exchanged between the systems.

  – If you have a disaster recovery plan that includes recovering to a hot site, you should consider how keys will be managed in this environment.

Documentation

Everything you do within the field of cryptography should be documented in detail. You will need it both in the initial startup, and later when you (or others) have to reconsider the current use of cryptography.

This documentation is extremely important. There is no such thing as trial and error in the cryptographic area to find out what was once done.

9.6 Initial Startup with SIU and PSCs

The SIU must be connected to the PS/2 (or PC) as shown in Figure 9-5 on page 9-13.

In your planning you have documented the cryptographic services you want to use, and the APIs that will be required. You will now use this information to configure the PSCs.

9.6.1 Configure the PSC Data Blocks

A PSC may be seen as a library where the data blocks are the physical files. Each data block serves a specific purpose. During configuration of the data blocks, different kinds of information is stored within each data block. In a later step this information is transferred to the AS/400.

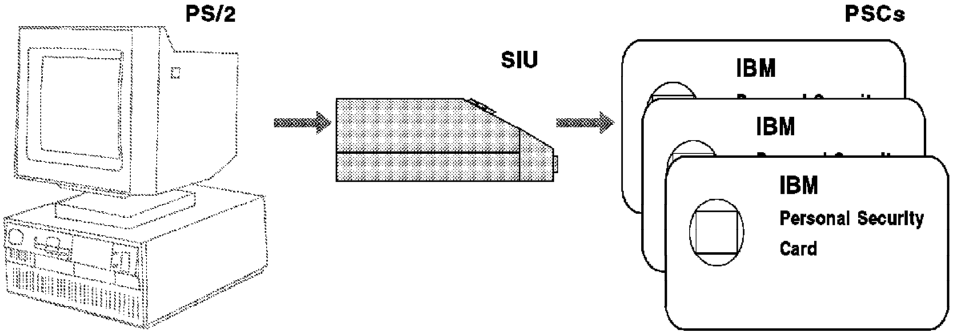


Figure 9-5. Using a PS/2 to Configure PSCs

The PS/2 must have the WSS program installed.

You must use *Workstation Security Services Installation and Operator′s Guide* to configure the PSC data blocks.

The following PSC data blocks are available:

1. ′CAKM1′ - Cryptographic Adapter Master Key part 1
2. ′CAKM2′ - Cryptographic Adapter Master Key part 2
3. ′PKAKM1′ - Public Key Algorithm Master Key part 1
4. ′PKAKM2′ - Public Key Algorithm Master Key part 2
5. ′SIUKM1′ - SIU Master Key part 1
6. ′SIUKM2′ - SIU Master Key part 2
7. ′CAXPSC1′ - Cryptographic Adapter to PSC ESS part 1
8. ′CAXPSC2′ - Cryptographic Adapter to PSC ESS part 2

   The ESS (Establish Secure Session) is used to establish a secure session between the cryptographic adapter in the AS/400 and the PSCs, via the SIU.
9. ′CAXPSC1′ - Cryptographic Adapter to SIU ESS part 1
10. ′CAXPSC2′ - Cryptographic Adapter to SIU ESS part 2
11. ′SIUXPSC1′ - SIU to PSC ESS part 1
12. ′SIUXPSC2′ - SIU to PSC ESS part 2
13. ′CAPROF0′ - Cryptographic Adapter profile 0 make file
14. ′CAPRFBC′ - Cryptographic Adapter public profile make file
15. ′CACFG′ - Cryptographic Adapter configuration data make file
16. ′SIUPROF′ - SIU profile make file
17. ′SIUCFG′ - SIU device configuration data make file
18. ′PROFVCT′ - Public Key Algorithm Profile Vector
19. ′CFGVCT′ - Public Key Algorithm Configuration Vector

9.6.2, “Initialize the Cryptographic Adapter (INZCS Command)” on page 9-14 tells where the PSC blocks are used.

The cryptographic profile determines which cryptographic services will be allowed. The cryptographic services are called using APIs in user-written programs.

The access a user can have to the cryptographic services are either NO ACCESS, or FULL ACCESS. There is no limited access, but you can control the access to certain services by:

- Restricting the cryptographic profile from specific cryptographic services APIs. This restriction overrules everything else. If you need certain cryptographic services not provided by the current cryptographic profile, you must activate another cryptographic profile that allows the services. You may secure the API that changes profiles by requesting a PIN to be keyed on the SIU.
- Limiting the authority to the corresponding APIs
- Limiting the authority to the programs that use the corresponding APIs, or using programs that adopt the owner′s authority.

There is only one API that takes exclusive control of the cryptographic adapter. It can be used to change a KEK, or to use the SIU to read data from the PSC to the AS/400.

There are four cryptographic profiles in the cryptographic adapter, two in the SIU and four in the PSCs. One of the cryptographic profiles in the cryptographic adapter must be enabled (activated). A cryptographic profiled is enabled using an API. You must decide if you want to secure this API by requiring the user to key in a PIN number whenever it is called. If you do, you must enter that PIN number at this stage.

9.6.2 Initialize the Cryptographic Adapter (INZCS Command)

The SIU must be disconnected from the PS/2 and connected to the cryptographic adapter in the AS/400 as shown in Figure 9-6 on page 9-15.

Authority to use the INZCS command should be granted only to specific users. The objects the user needs authority to are as follows:

- INZCS command in QSYS
- QC3INZ program in QTSS

The purpose of the command INZCS with INZOPT(*DES) is to initialize the cryptographic processor:

- Download the microcode to the cryptographic processor
- Initialize the cryptographic adapter with:
  - The master key
  - The ESS keys
  - Profile make files
  - Configuration data make files

All the information from the PSCs will be transferred to the SIU and the cryptographic adapter in the AS/400 when the INZCS command is run.

The date and the time in the IBM Security Interface Unit is set from the AS/400 system clock when the INZCS command is run. The IBM Security Interface Unit provides the time to the IBM Cryptographic Adapter and IBM Personal Security Card during a secure session. This date and time may be changed using an API.

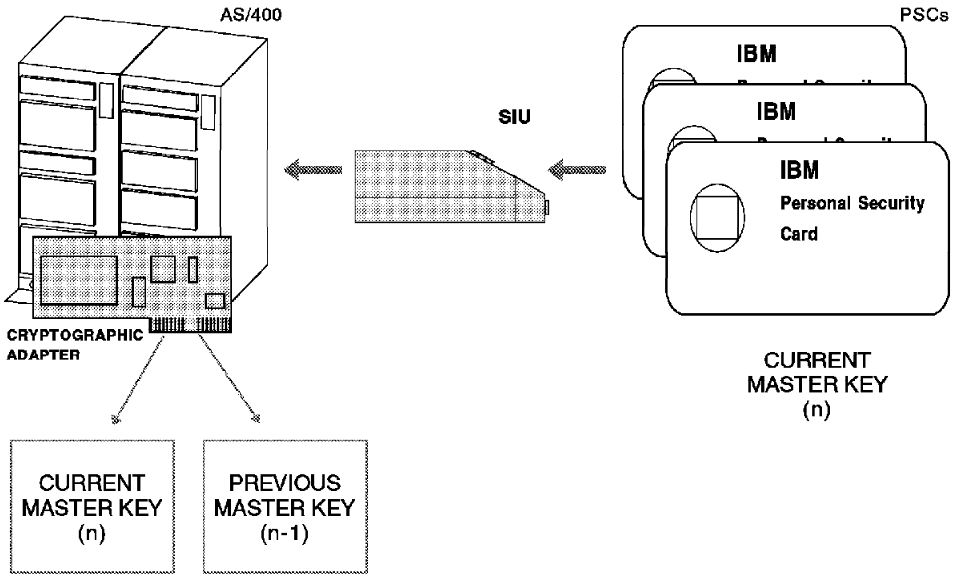


Figure 9-6. Transport of Data from PSCs to the AS/400

The following PSC data blocks are transferred from the PSC to the SIU when issuing the INZCS command:

- ′SIUKM1′ - SIU Master Key part 1
- ′SIUKM2′ - SIU Master Key part 2
- ′SIUXPSC1′ - SIU to PSC ESS part 1
- ′SIUXPSC2′ - SIU to PSC ESS part 2
- ′SIUPROF′ - SIU profile make file
- ′SIUCFG′ - SIU device configuration data make file

The following PSC data blocks are transferred from the PSC to the SIU and the cryptographic adapter in the AS/400:

- ′CAXSIU1′ Cryptographic Adapter to SIU ESS part 1
- ′CAXSIU2′ Cryptographic Adapter to SIU ESS part 2

The following PSC data blocks are transferred from the PSC to the cryptographic adapter in the AS/400:

- ′CAKM1′ - Cryptographic Adapter Master Key part 1
- ′CAKM2′ - Cryptographic Adapter Master Key part 2
- ′PKAKM1′ - Public Key Algorithm Master Key part 1
- ′PKAKM2′ - Public Key Algorithm Master Key part 2
- ′CAXPSC1′ - Cryptographic Adapter to PSC ESS part 1
- ′CAXPSC2′ - Cryptographic Adapter to PSC ESS part 2
- ′CAPROF0′ - Cryptographic Adapter profile 0 make file
- ′CAPRFBC′ - Cryptographic Adapter public profile make file
- ′CACFG′ - Cryptographic Adapter configuration data make file
- ′PROFVCT′ - Public Key Algorithm Profile Vector

- ′CFGVCT′ - Public Key Algorithm Configuration Vector

Things to Remember about INZCS

- The system must be in restricted state.
- Save the master key version number with the master key.

  The master key has a master key version number (MKVN) that increases by one every time INZCS is run. In V3R1 the MKVN for DES master key is stored in QAC4KEYST in QUSRSYS, and the PMKVN for the PKA master key is stored in QAC4PKEYST in QUSRSYS.
- Do not run INZCS twice without STRCS in between.

  If INZCS is run twice without running STRCS in between, the master key is changed twice. If the key storage is not re-enciphered all the keys in key storage will be lost. See Figure 9-7 on page 9-17.
- Check the status of re-encipher batch job. Do not run ENDCS before the re-encipher batch job has completed.
- Do not exit INZCS during data block reads.
- Make sure all cards are available before starting INZCS.
- Make sure that at least one profile is authorized to read data blocks.

Refer to *Common Cryptographic Architecture Services/400 Installation and Operator′s Guide* for more information about INZCS.

9.7 Start of the Cryptographic Support (STRCS Command)

The purpose of the STRCS command is to start the cryptographic processor so it can accept requests from cryptographic APIs.

Authority to use the STRCS command should be granted only to specific users. The objects the user needs authority to are:

- STRCS command in QSYS library
- QC3INZ program in QTSS library
- QC3RTCMK program in QTSS library
- QSYSNOMAX job queue in QSYS library

If REENCIPHER is set to *YES in the STRCS command two batch jobs are submitted that re-enciphers all the keys in QAC4KEYST and QAC4PKEYST using the new master key. (QAC4KEYST and QAC4PKEYST are often referred to as key storage).

Re-enciphering of a key means that the key is brought into the cryptographic adapter in plaintext using the current master key, and enciphered under the new master key. It does not change the content of the key, so any data encrypted under the old key can be accessed by the new re-enciphered key.

You may for example have a backup of data that is encrypted under OLDKEY. OLDKEY is re-enciphered and is now NEWKEY. You can restore your backup and use NEWKEY as long as the plaintext of NEWKEY is not changed. See Figure 9-7 on page 9-17.

If another system accesses encrypted data on your system, for example, using CPI-C or DDM, they do not have to do anything as long as you do not change the key′s plaintext.

If you have programs that store keys in files other than QAC4KEYST or QAC4PKEYST you must re-encipher these keys before the master key is changed again. If not, the keys and all the data enciphered under them will be lost.

IMPORTANT!

You must be in complete control of INZCS, STRCS, and re-enciphering of keys.

Beware of the following situations:

1. STRCS submits a batch job that re-enciphers the keys in the key storage.

   This may take a few minutes to complete depending on the size of the key storage and the system load.

2. ENDCS is done while the batch job is still running.

   The batch job is still trying to re-encipher the remaining keys in the key storage, so if STRCS is stopped the rest of the keys will not be re-enciphered.

   - To have the remaining keys correctly re-enciphered another STRCS with REENCIPHER(*YES) must be run before a new INZCS is run. If a new INZCS is run, the keys in key storage that had not been re-enciphered at the time of ENDCS are lost.

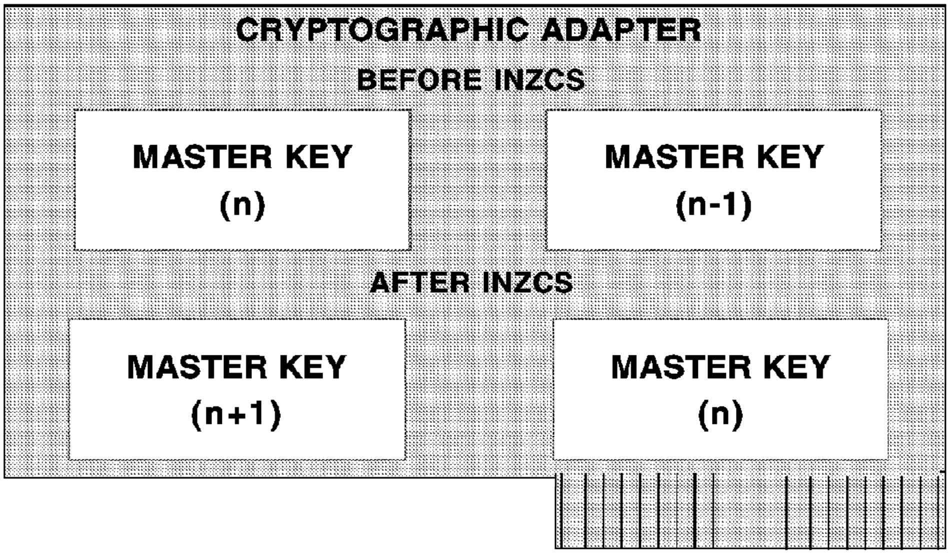


Figure 9-7. Change of the Master Keys within the Cryptographic Adapter

Each time the INZCS is run, the master key and the MKVN are changed, and the old master key is moved to a special storage within the cryptographic adapter. In Figure 9-7 master key (n + 1) represents a new master key. The current master key (n) becomes the old master key when INZCS is run.

The keys in key storage are automatically re-enciphered under the new master key (n + 1) by the batch job started by STRCS. If your applications have keys stored outside the key storage they are still enciphered under the old master key (n). It is your responsibility to re-encipher them under the new master key (n + 1). This is automatically done when your APIs call them, or by using the Key_Token_Change SAPI verb. The old master key (n) is used to decipher the keys, and the new one (n + 1) is used to encipher them again.

If you run INZCS twice without re-enciphering your keys their old master key (n - 1) will be gone. The keys can not be deciphered and are lost. When you lose a key you also lose the data encrypted under it.

If RESET is set to *YES in the STRCS command:

- The microcode in the cryptographic processor is reloaded
- The microcode in the cryptographic adapter is **NOT** reloaded
- The master keys (both DES and PKA) are **NOT** reset

Refer to *Common Cryptographic Architecture Services/400 Installation and Operator′s Guide* for more information about STRCS and how to back up and recover keys.

9.8 Keys

CCAS/400 can use a variety of keys to request cryptographic services. Keys are separated into types by associating a key with a non-secret value, called a control vector, that contains information about permitted use of the key.

In the basic cryptographic process a key is brought into the cryptographic adapter from its storage (such as QAC4KEYST in library QUSRSYS) and used on the data to be transformed from plaintext to ciphertext, or vice versa.

Using different keys on the same data produces different results.

The master key is the only key that is loaded into the system. All other keys are created using APIs in user written programs.

If the master key for some reason must be transmitted to another system you should use PSCs.

The different key types are:

- Master key - Stored in clear in the cryptographic adapter.
- Key-encryption keys (KEK) - Used to encipher keys sent to other systems, or decipher keys received from other systems.
- PIN keys - Used to generate and verify PINs.
- Cipher keys - Used to cipher or decipher data.
- MAC keys - Used to generate MACs.
- MDC keys - Used to generate MDCs.
- Compatibility keys - Used to cipher and decipher data, and to generate MACs.

- PKA keys - Used to transmit and receive enciphered keys between nodes in a network.

Refer to *Common Cryptographic Architecture Services/400 Installation and Operator's Guide* for more information about keys, their hierarchy, their different forms, and control vectors.

9.9 Example of How Cryptography Works

In this example we will look at the process of performing simple encryption and decryption of data being sent between systems. Figure 9-8 provides an outline of the environment:

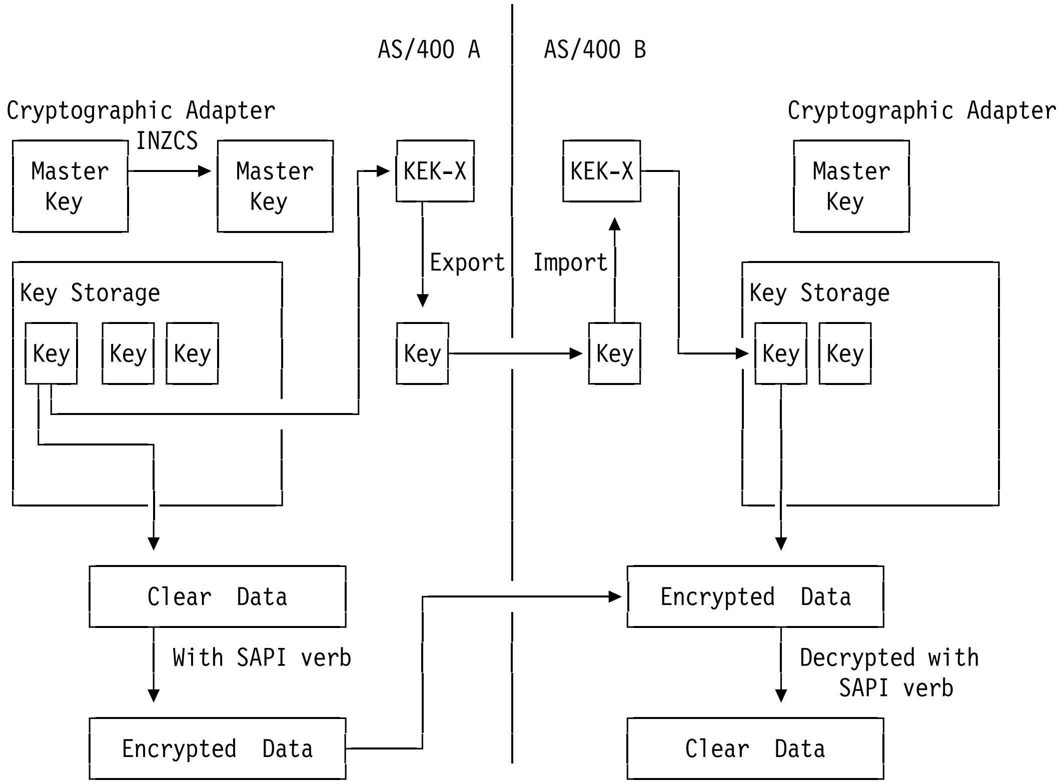


Figure 9-8. Example of Encryption/Decryption Environment

Every system participating in the cryptographic environment will have a master key. On the AS/400, the master key is set with the INZCS CL command. The INZCS command also allows you to change the master key. This command can only be run when the system is in restricted state. In this way, operations staff are limited to changing the master key.

The master key is stored in the cryptographic adapter in a shielded and protected area. If this area is tampered with, the master key is destroyed.

Users can create multiple cryptographic keys for encrypting and decrypting data. These are stored in an encrypted form in a key storage object on the AS/400. The master key is used to encrypt these keys. If the master key changes, all of the keys in the key storage are re-encrypted using the new master key.

To encrypt data, a SAPI verb is used in a program, using the cryptographic key which works on the clear data, creating the encrypted data. This encrypted data can then be sent to another system via a communications line, or be saved to tape and transported by hand.

On the receiving system, the corresponding cryptographic key must also be available to decrypt the data. To set up the second system (AS/400 B in our example), the cryptographic key must be *exported* using a key encryption key (KEK). This exported cryptographic key can then be transported to system B using a personal security card, and imported using a key encryption key on that system, to create the equivalent cryptographic key. This cryptographic key is stored in the key storage object, encrypted using the master key on that system.

A program on system B can read that encrypted data, and, using a SAPI verb, use the cryptographic key to decrypt the data into clear form.

9.10 Public Key Algorithm (PKA)

PKAs are used to:

- Encrypt DES key-encryption keys (KEKs) that are sent between nodes in a network.
- Generate and verify digital signatures that are associated with public keys or DES KEKs that are sent between nodes in a network.

PKA provides a solution to the DEA key distribution problem. Instead of using one secret key for encryption and decryption, PKA uses two keys, one public key and one private key. The two keys are generated as a pair. The operation that uses the public key is normally referred to as encryption while the operation that uses the private key is referred to as decryption, but either key may be used first. To return the data to the original form, the other must be used.

The difference between DES keys and PKA keys is that DES keys require the same key to encipher and decipher, while the PKA keys use one key to encipher and another to decipher. DES keys are referred to as *symmetric keys*, PKA keys as *asymmetric keys*.

When two users wish to establish a secure channel between two systems, both send their public key to the other over an unsecured channel. When one user enciphers data using the other user′s public key, only the other user′s private key can be used to decipher the data. The private keys remain on the system where they were created. They are secret and are not sent anywhere.

The main objectives of the public-key services are to distribute DES keys securely and to generate and verify digital signatures. The DES services perform general encryption and decryption. The public-key services do not replace or modify the DES services.

PKA has its own master key, and it is treated just like the master key for DES.

9.11 Other Cryptographic Facilities Available on the AS/400

Scramble QUSRTOOL: Scramble is a program that rearranges the letters in a text file. For example the word COMPUTER could be scrambled to be PTMECORU. You can only scramble text, not binary or decimal data. Scramble is based on secure keys. It does not provide any methods for key management. Scramble is very easy to use, but does not give good protection. Scramble does not have any export restrictions.

Cryptographic Support/400 Software (5738-CR1): Cryptographic Support/400 provides security for data stored on tape, or diskette, or in the system database. Both communication and file security procedures are similar. Cryptographic Support/400 is effectively superseded by the hardware cryptographic functions offered with IBM Common Cryptographic Architecture Services/400.

Cryptographic Support/400 offers the following three functions:

1. Data Encryption and Decryption

   Change plaintext into ciphertext or vice versa using Data Encrypting Algorithm (DEA). Data encryption and decryption are also used to generate MACs.

2. Key Management

   Protects cryptographic key identities through the use of:

   - Data encrypting key management, including key generation, key encryption and key translation
   - Cross-domain key management, including maintenance of a cross-domain key table
   - Installing a host system master key

3. Personal Identification Number (PIN) Processing.

   Three PIN functions are supported:

   - Generation
   - Validation
   - Translation

The U.S. export restrictions apply to the Cryptographic Support/400. See “Export Restrictions for DES” on page 9-2 for more information about the export restrictions.

Table 9-2 on page 9-22 compares the three cryptographic tools available on the AS/400.

Table 9-2. Comparison of Cryptographic Features

Cryptographic Function	IBM Common Cryptographic Architecture Services/400 PRPQ	IBM Cryptographic Support/400 Software (5738-CR1)	Scramble QUSRTOOL	Function and details
Where are the cryptographic functions performed	Hardware	Software	Software	
Encipher and decipher	Yes	Yes	No	Scramble just changes the order of the letters in a text.
MAC Generate	Yes	Yes	No	
MAC Verify	Yes	No	No	
Personal Identification Numbers	Yes	Yes	No	CCAS provides most PIN functions.
Max. data length per operation	16777215	65527	256	Scramble does not allow chaining
Key Management	Yes	Yes	No	
PKA	Yes	No	No	
Link Level Encryption	No	No	No	
Session Level Encryption	No	No	No	
Signature Verification	No*	No	No	* Supported on TSS workstations only

Chapter 10. Communications Security in SNA

This chapter tells you about IBM′s *System Network Architecture (SNA)* and how it relates to AS/400. We will also define some of the terms that are used when AS/400 is part of an SNA network.

This chapter will also show you how to define the different communications descriptions on the AS/400 and how security can be integrated into these definitions.

If AS/400 communications and SNA is new to you, then this chapter will show you how SNA and AS/400 work management contribute to AS/400 security. Security in IBM supplied communications applications is covered in Chapter 11, “Security in IBM Communications Applications.”

10.1 Architectures

A network architect will design a network given several key pieces of information, such as:

- System location(s) - existing and planned sites
- End-user profiles - facilities required for the users of the system
- Traffic - data volumes between locations
- Tariffs - to take advantage of the most economical telecommunication provider offerings
- Costs - what type of lines, connections, and speeds will be used: leased, switched, V.35, V.24, X.25, X.21, ISDN - there are multiple options
- Availability and reliability of the network

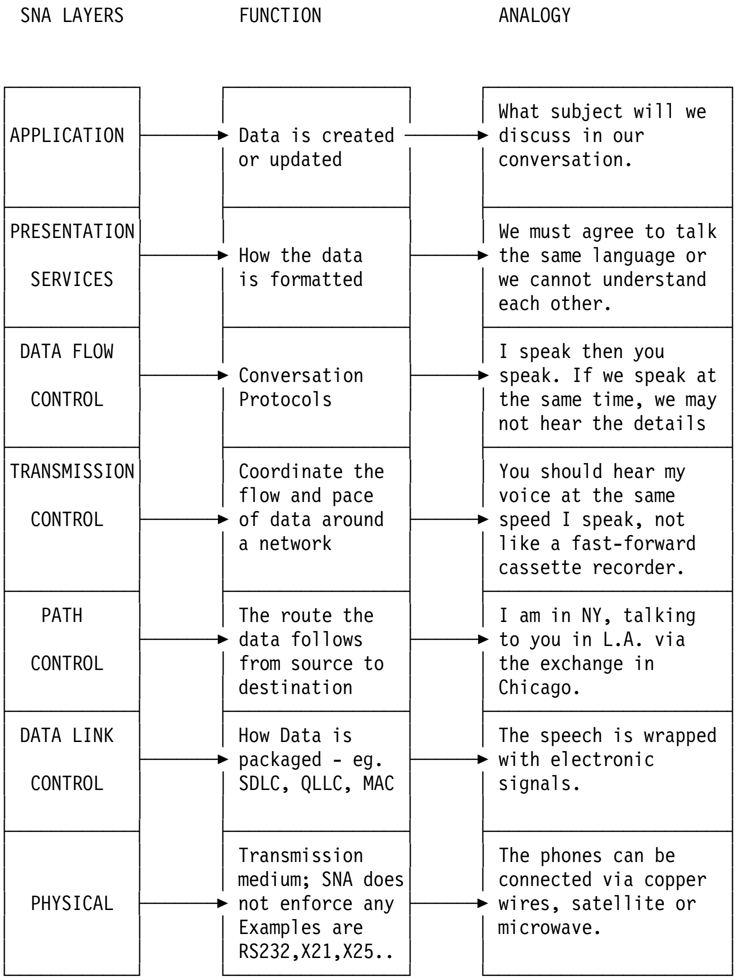
Security should also be included as part of the network design. Physical measures may be implemented to make it harder to get to the data flowing through the network, for example, limiting access to site wiring closets, and enclosing communications cables in tamper-proof conduits. However, this cannot ensure that the user accessing the system using communication lines is legally entitled to do so. User access can be controlled by also implementing some degree of security in the *communications architecture*.

A communications architecture defines the rules for how data is sent across a network. This should be designed for maximum flexibility so that a network may grow, and not be prevented from incorporating and taking advantage of developments in technology. Examples of communications architectures are TCP/IP (Transmission Control Protocol/Internet Protocol), used extensively in UNIX networks, SNA (Systems Network Architecture), IBM′s implementation of a communications architecture, and OSI (Open Systems Interconnection), an evolving architecture for compatibility across computer manufacturers.

10.1.1 Systems Network Architecture

SNA is IBM′s architecture for communications systems. It is a ″blueprint″ of the logical structures, formats, protocols, and operational sequences, for transmitting information through networks.

These definitions or rules are grouped into seven layers. Each layer covers separately defined functions in a network. For example layer 3, which is defined as Path Control, controls how data can be routed from one location in the network to another. Figure 10-1 provides a simple explanation and analogy for the component layers of SNA.



SNA LAYERS	FUNCTION	ANALOGY
APPLICATION	Data is created or updated	What subject will we discuss in our conversation.
PRESENTATION SERVICES	How the data is formatted	We must agree to talk the same language or we cannot understand each other.
DATA FLOW CONTROL	Conversation Protocols	I speak then you speak. If we speak at the same time, we may not hear the details
TRANSMISSION CONTROL	Coordinate the flow and pace of data around a network	You should hear my voice at the same speed I speak, not like a fast-forward cassette recorder.
PATH CONTROL	The route the data follows from source to destination	I am in NY, talking to you in L.A. via the exchange in Chicago.
DATA LINK CONTROL	How Data is packaged - eg. SDLC, QLLC, MAC	The speech is wrapped with electronic signals.
PHYSICAL	Transmission medium; SNA does not enforce any Examples are RS232,X21,X25..	The phones can be connected via copper wires, satellite or microwave.

Figure 10-1. SNA Layers - Function and Analogy

The layered structure of SNA provides flexibility and separation of functions. Changes to features implemented by one layer should not impact the functions of other layers. For example, the Inventory application should continue to run unaltered when site wiring is converted from twinax cables to token-ring.

SNA does not *enforce* security. Rather, it is an architecture that allows for security implementation. We will see later that the AS/400 uses these features in a variety of ways, most appropriate for the particular application. SNA also allows for robustness and integrity of data in a communication environment.

Note: OSI also operates with a structure of seven layers.

10.2 SNA Communications Security

Security in an AS/400 communications environment is different from resource and object security on a single system.

If you are auditing communications security you need to know about AS/400 communications configuration to be able to see what systems and functions users have access to.

You may want to read this section if AS/400 communications configuration is new to you, or if you want to be updated on how security can be part of your AS/400 configuration.

10.2.1 Introduction

This section talks about the different configuration options and parameters you have when connecting AS/400 to a network, and how they relate to security. If you are auditing AS/400 communications security, it is important for you to know what the options are, and how and where they are specified.

In this section we also look at how communications configuration on the AS/400 works to prevent ″invalid″ users access to the system. We will distinguish between two types of security implementation: non-LU 6.2 and LU 6.2.

As mentioned previously, SNA does not *enforce* security. Rather, it provides a framework for security. SNA security is enabled by certain configuration parameters and values. These relate to specific SNA layers, although the implementer doesn′t have to be aware of this.

10.2.2 Configuring AS/400 Communications

The AS/400 configuration is object oriented, and several configuration objects can be defined:

- Network Interface Description (NWID)
- Line Description (LIND)
- Controller Description (CTLD)
- Device Description (DEVD)
- Mode Description (MODD)
- Class of Service Description (COSD)

Collectively these define the necessary SNA components for AS/400 communications.

Configuration descriptions are linked together in a hierarchy. In the above list, each level is referenced by the previous level. In addition, NWIDs can have many LINDs defined, LINDs can have many CTLDs attached, and each CTLD can have many DEVDs attached.

Further AS/400 components involved in communications are Configuration lists (discussed in 10.3.2, “Session Level Security” on page 10-14) and Subsystems (discussed in 1-8 and in 10-10).

Network Interface Description (NWID)

NWID is used in ISDN networks. An integrated services digital network (ISDN) is a public or private digital communications network that supports multiple channels of voice, data, image, and other services over the same physical interface.

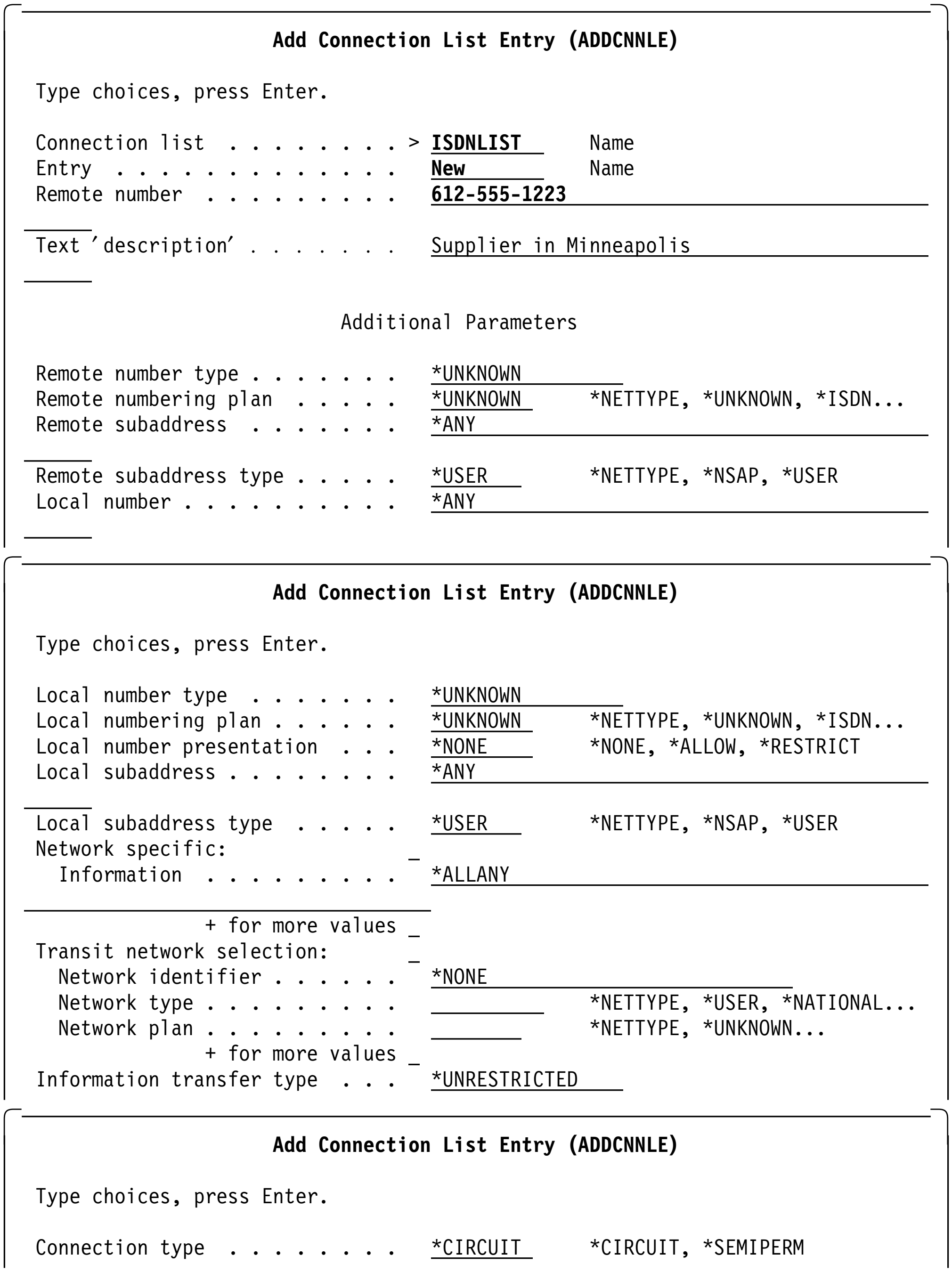
An ISDN connection requires a *Network Interface Description* (NWID) and, for some connection types, a *Connection List*.

A Network Interface Description is a system object that represents the AS/400 physical interface to ISDN. Each network interface description has a physical resource name associated with it.

The network interface description allows two B-channels for data transport and a D-channel to exchange signals with the network. The line descriptions represent the logical connections (B-channels) in a physical interface.

The ISDN Connection List: Provides information on when to accept incoming calls and what information to send with an outgoing call. This information is like a list of telephone numbers and is used only for switched connections. A default connection list (QDCCNNLANY) is provided with the system and is configured for answering incoming ISDN calls. The Change Network Attributes (CHGNETA) command can change the default connection list used to accept incoming calls.

Figure 10-2 on page 10-5 shows the Add Connection List Entry (ADDCNNLE) command prompt screen.



```
                    Add Connection List Entry (ADDCNNLE)

 Type choices, press Enter.

 Connection list . . . . . . . . > ISDNLIST      Name
 Entry  . . . . . . . . . . . . .  New           Name
 Remote number  . . . . . . . . .  612-555-1223

 Text 'description' . . . . . . .  Supplier in Minneapolis


                            Additional Parameters

 Remote number type . . . . . . .  *UNKNOWN
 Remote numbering plan  . . . . .  *UNKNOWN      *NETTYPE, *UNKNOWN, *ISDN...
 Remote subaddress  . . . . . . .  *ANY

 Remote subaddress type . . . . .  *USER         *NETTYPE, *NSAP, *USER
 Local number . . . . . . . . . .  *ANY
```

```
                    Add Connection List Entry (ADDCNNLE)

 Type choices, press Enter.

 Local number type  . . . . . . .  *UNKNOWN
 Local numbering plan . . . . . .  *UNKNOWN      *NETTYPE, *UNKNOWN, *ISDN...
 Local number presentation  . . .  *NONE         *NONE, *ALLOW, *RESTRICT
 Local subaddress . . . . . . . .  *ANY

 Local subaddress type  . . . . .  *USER         *NETTYPE, *NSAP, *USER
 Network specific:                _
   Information  . . . . . . . . .  *ALLANY

                + for more values _
 Transit network selection:       _
   Network identifier . . . . . .  *NONE
   Network type . . . . . . . . .                *NETTYPE, *USER, *NATIONAL...
   Network plan . . . . . . . . .                *NETTYPE, *UNKNOWN...
                + for more values _
 Information transfer type  . . .  *UNRESTRICTED
```

```
                    Add Connection List Entry (ADDCNNLE)

 Type choices, press Enter.

 Connection type  . . . . . . . .  *CIRCUIT      *CIRCUIT, *SEMIPERM
```

Figure 10-2. The Add Connection List Entry (ADDCNNLE) Command.

The Line Description specifies the Connection List that incoming calls should use. This is specified with the CNNLSTIN parameter. This parameter defaults to *NETATR.

Line Description

The *Line Description* (LIND) defines the way the physical port is used. Here, data link protocol (SDLC, TRN, and so on) and the physical and logical parameters that condition the protocol's behavior are defined.

LINDs are created using the appropriate CRTLINxxxx command. See the *OS/400 Communications Configuration Reference* for a full description. A *SECURITY* parameter can be defined for token-ring, SDLC, and X.25 line descriptions, but it is only valid if APPN is used on the system. Together with the COS the SECURITY parameter is used to determine the route selection through an APPN network. Refer to the redbook *Centralized Security Administration in an APPN Network*, GG24-3719, for more information. Other parameters specified on the LIND are used during partner identification at session establishment. The values are shown in Table 10-1.

Table 10-1. Security Related Parameters on the Line Descriptions. The SECURITY parameter applies only to token-ring, SDLC and X.25 links and is used only for lines attached to APPC or host controllers using APPN. It is used by APPN for Class-of-Service processing.

LIND Parameter	*Value*	*Meaning*
SECURITY	*NONSECURE	No security on the line
	*PCKTSWNET	The line is secure in that there is no fixed route for the data packets
	*UNDGRDCBL	This is an underground cable (considered secure)
	*SECURECND	A secure conduit but not guarded (for example, a pressurized pipe)
	*GUARDCND	The line is a guarded conduit protected against tapping
	*ENCRYPTED	Data flowing on the line is encrypted
	*MAX	Guarded conduit protected against physical and radiation tapping
EXCHID	AS/400 value is always 056nnnnn	Identifies the local AS/400 Exchange ID (XID) is composed of block number (product specific - 056 for AS/400) and ID number (nnnnn) chosen by the installation. It uniquely identifies a specific station in the network Usually required for switched SDLC networks; optional for other line types If specified, it will be exchanged with remote system If XIDs do not match, the session is not established
ADPTADR	If the resource name is *NWID, the explicit address must be specified	The AS/400 token-ring or Ethernet adapter address
STNADR	2 hexadecimal digits	Station Address Used for BSC multipoint lines for the remote system to poll the AS/400 Used for SDLC lines answering on a switched line Valid for AS/400 as non-primary role; switched lines, SNBU

Note: The Security parameter values simply *describe* the security characteristics of the line; this is in contrast to the host NCP/VTAM specifications, which *determine* the characteristics for the line.

Controller Description

The *Controller Description* (CTLD) defines the characteristics of the remote system (an ES/9000 host, S/36, S/38, DOS, OS2, RS/6000, a non-IBM system, or another AS/400). You configure the CTLD using one of the CRTCTLxxxx commands. (See the *AS/400 Programming: Control Language Reference* for details). Different types of CTLDs implement security in different ways. Security related parameters for the CTLDs are shown in Table 10-2.

CTLD Parameter	*Value*	*Meaning*
RMTNETID	*NETATR, *NONE, remote-network-id	RMTNETID (remote network identifier) and RMTCPNAME (remote control point name) of the attaching system. Both are required for APPN. If not specified for host connection - SSCPID must be used.
RMTCPNAME	remote-cp-name	
EXCHID	AS/400 value is always 056nnnnn	Identifies the remote controller Exchange ID (XID) is composed of block number (product specific - 056 for AS/400) and ID number (nnnnn) chosen by the installation It uniquely identifies a specific system in the network Usually required for switched SDLC networks - optional for other line types If specified, it will be exchanged with the remote system If exchange IDs do not match, the session is not established .
SSCPID	000000000001 thru FFFFFFFFFFFF	System Services Control Point ID Used for Finance, Remote Workstation, and Retail Controllers Usually used for switched SDLC lines It will be exchanged with the remote system, if specified If they do not match, the session is not established

Table 10-2. Security Related Parameters in the Controller Description.

Device Description

The *Device Description* (DEVD) defines the physical or logical device to which sessions will be established, for example, a physical screen or printer. A remote communications resource is represented on the AS/400 as one or more DEVDs. Security related parameters specified on the DEVD are shown in Table 10-3.

DEVD Parameter	*Value*	*Meaning*
RMTLOCNAME	Remote location name	Required on most communication DEVDs Must match local location name on remote system configuration
LCLLOCNAME	Local location name *NETATR (the value in the network attributes)	The name by which the local AS/400 is known to other systems in the network Each location must have a unique name in the network Must match remote location name on the remote system configuration
RMTNETID	Remote Network Identifier *NETATR (the value in the network attributes)	The name of the remote network This must match the Local NETID network attribute on the remote system
SECURELOC	*YES *NO	Determines if the remote system can be trusted as a target (can specify SECURITY(SAME)) Used by APPC devices Not used if APPN (*YES) and LOCADR(00) are specified Ignored if system security level is set to 10
LOCPWD	*NONE Location-password (hex)	Determines bind level security Valid for APPN (*NO) Valid for APPN (*YES) when DEVD manually configured This information is in the APPN remote configuration list for APPN(*YES) Ignored if system security level is set to 10

Table 10-3. Security Related Parameters in the Device Description. LOCPWD is the location password that can be used to specify security at the bind (session establishment). SECURELOC (secure location) is used to specify conversation level security. Refer to the text for a discussion of bind and conversation levels of security (LOCPWD and SECURELOC).

Not all communications devices make use of all the DEVD security parameters. These are summarized in Table 10-4.

*Table 10-4. Security Related Parameters in Device Descriptions by Device Type. The table shows which DEVD parameters are required ("R") to be specified for the given device type, or default ("D") to *NETATR or some other value.*

DEVD Parameter	*Device description type: CRTDEVxxxx*											
	APPC	*ASC*	*BSC*	*DSP*	*FNC*	*HOST*	*INTR*	*NET*	*PRT*	*RTL*	*SNUF*	*SNPT*
RMTLOCNAME	R	R	R		R(4)	R	R		R(1)	R	R	
LCLLOCNAME	D								R(1)			
RMTNETID	D								R(1)			
LOCADR	D		R	R(3)	R	R(5)			R(3)	R	R	R
LOCPWD	D(2)											
SECURELOC	D(2)											

Note:

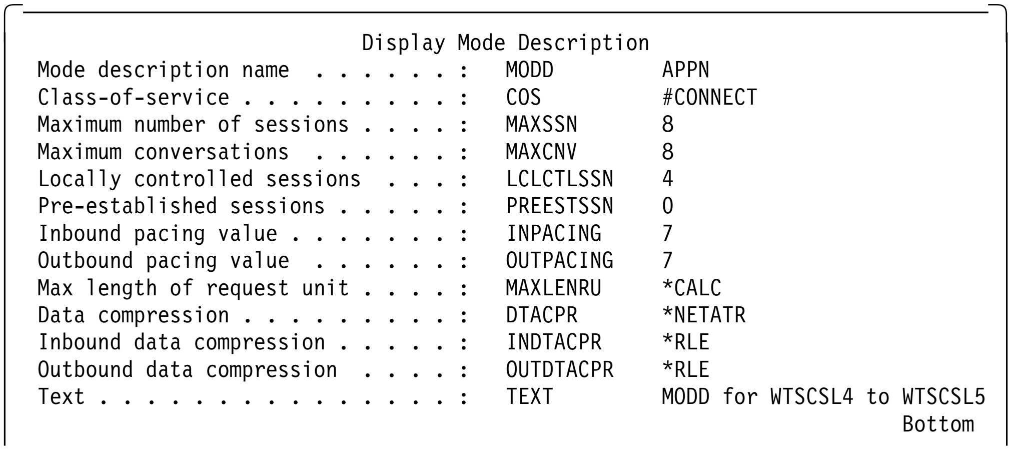
1. For advanced function printers (AFP) only
2. Not valid where APPN(*YES) and LOCADR(00)
3. For remote devices (DEVCLS(*RMT)) only
4. Cannot be specified for any type except *FNCICF
5. Must match LOCADDR of NCP LU definition

Mode Descriptions

The *Mode Description* (MODD) defines the data flow control (layer 5) parameters for a session and is used only for APPC (LU 6.2) and APPN. A MODD identifies the characteristics and number of sessions for a logical unit. An example of a MODD is shown in Figure 10-3.

A MODD can be used to limit the number of sessions (MAXSSN) and conversations (MAXCNV) that exist between a pair of locations. Although this is not its prime purpose, it can provide additional security for APPC applications, by forcing an application to use a defined mode.

If problems occur which warrant the termination of certain communications activities, the end mode (ENDMOD) command can be used. The group of sessions sharing that mode will end without interrupting other users sharing the same physical communications line.



```
                             Display Mode Description
Mode description name  . . . . . . :   MODD         APPN
Class-of-service . . . . . . . . . :   COS          #CONNECT
Maximum number of sessions . . . . :   MAXSSN       8
Maximum conversations  . . . . . . :   MAXCNV       8
Locally controlled sessions  . . . :   LCLCTLSSN    4
Pre-established sessions . . . . . :   PREESTSSN    0
Inbound pacing value . . . . . . . :   INPACING     7
Outbound pacing value  . . . . . . :   OUTPACING    7
Max length of request unit . . . . :   MAXLENRU     *CALC
Data compression . . . . . . . . . :   DTACPR       *NETATR
Inbound data compression . . . . . :   INDTACPR     *RLE
Outbound data compression  . . . . :   OUTDTACPR    *RLE
Text . . . . . . . . . . . . . . . :   TEXT         MODD for WTSCSL4 to WTSCSL5
                                                                        Bottom
```

Figure 10-3. Example of a Mode Description (MODD)

Class of Service

The *Class of Service* (COS) defines the path control (layer 3) parameters and is applicable only to APPN. It manages data flow in a network, for example, to take advantage of the least congested routes, the highest speed, and the lowest tariffs.

COS offers the capability of choosing routes through secure paths as shown with the SECURITY parameter in Table 10-1 on page 10-6.

Remote Location Name

Remote Location Name is used to make application programs independent of the physical communication device. A remote location name is an SNA Logical Unit (LU) name that is used to select which particular DEVD is used.

Each AS/400 can have a single Remote Location Configuration List. It is created with the Create Configuration List (CRTCFGL) command. It contains a list of remote locations, their location password, and whether or not the remote location is secure. It is needed if you want a LOCPWD or SECURELOC other than the defaults, or if you want to route through non-APPN networks, for example VTAM. A description of secure/non-secure locations is given in 10.3.2, “Session Level Security” on page 10-14.

The selection of the DEVD determines the physical link that the application program has with the remote system. This does not take place until the application program needs to communicate. Since the remote location name is used to determine the *type* of communications that will occur, different types of DEVD cannot have the same remote location name. The system searches for a DEVD with the required remote location name. When one is found the corresponding controller and its line description is found too.

Note: When **APPN(*NO)** is used the devices are searched alphabetically for *all* the descriptions that match the remote location name, the local location name, and the remote network ID parameters. The system chooses the first device that matches these parameters, and which is in a state that can be used.

When **APPN(*YES)** is used the system determines a route to the remote location, and searches for a device with the remote location within the specified remote network ID. The device selected must be attached to the controller description representing the first connection of the calculated route. If a device is selected, it is varied on if necessary. If a device does not exist, one is automatically created and activated.

Subsystems

In “Subsystems” on page 1-8, we introduced the concept of AS/400 work entries. A *Communications Entry* is an example of a work entry. It is included in a subsystem by the ADDCMNE (add communications entry) command.

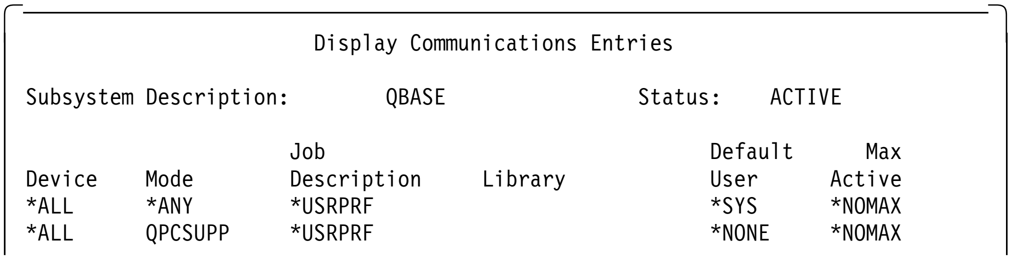
Communications devices (DEVDs) are simply a source of work for a subsystem. Each communications entry defines one or more devices or remote locations that are controlled by the subsystem. The devices are allocated by the subsystem for receiving program start requests for the communications jobs.

To view communications entries for a given subsystem use the command:

```
DSPSBSD SBSD(name-of-subsystem)
```

and choose option 8 (Communications entries) or option 9 (Remote Location Name entries) at the next screen.

A subsystem′s communications entry is an important security feature. For example, when the AS/400 is shipped, QBASE has predefined communications entries as shown in Figure 10-4.



```
                        Display Communications Entries

Subsystem Description:       QBASE              Status:    ACTIVE

                      Job                                  Default      Max
Device    Mode        Description     Library              User       Active
*ALL      *ANY        *USRPRF                              *SYS       *NOMAX
*ALL      QPCSUPP     *USRPRF                              *NONE      *NOMAX
```

Figure 10-4. Default Communications Subsystem Entries in QBASE.

These default entries keep communications configuration tasks to a minimum. However, it allows communications (evoke) requests from *any* source to be processed through this entry, but only if required security information is present.

The default user *NONE parameter means that no user profile is specified as default, which means that requests without a supplied user profile will be rejected.

The default user *SYS means that all user program start requests will be treated as *NONE. For program start requests sent by system functions, the request will run under a predetermined user profile if a user profile is not specified on the program start request.

10.2.3 Base Level SNA Security

During session establishment, there is an exchange of information between the systems trying to establish a session. For example, on switched SDLC lines, the EXCHID defined in the line and controller descriptions are sent between the systems. If the target and source systems are able to identify the EXCHID received it will be validated and a session can be established. This is a security implementation at the data link control layer of SNA.

Once the session has been established, the LUs allow the sending of user profile and password for the users to identify themselves to the resource. A user on the source system cannot sign on to the target system without having a valid user profile and password. Accordingly, effective user profile and password management techniques are necessary in order to protect the target AS/400. User profile considerations are covered in Chapter 3, “User Profiles and Group Profiles” on page 3-1 and passwords are covered in 3.1.1, “Password” on page 3-2.

10.3 APPC and APPN

To understand APPC and APPN, there are some basic terms that you should be familiar with. The following topics are the most important ones.

Logical Unit (LU):

- An entry point into an SNA network, mainly for application programs (LU0 and LU6.2) or devices (LU0, 1, 2, 3, 4, 7)
- For APPN(*NO) it is an AS/400 APPC device description
- For APPN(*YES) it is an AS/400 device with remote location name (RMTLOCNAME), local location name (LCLLOCNAME), and remote network identifier

Session: A connection between two LUs. APPC can support multiple (parallel) sessions between the same LUs.

Conversation: A connection between two programs across a session. A session can be serially reused (that is, one at a time) by different conversations.

Advanced Program-to-Program Communications (APPC): APPC (also known as LU 6.2 and used synonymously) was the first implementation of the PU (Physical Unit) type 2.1 (point-to-point) communications architecture.

Note: In the strict definition, LU 6.2 is the type of logical unit that is capable of supporting the Advanced Program-to-Program Communications function.

LU 6.2 provides connection between transaction programs and the network resources. Each LU makes a set of resources available to its transaction programs. These resources vary, depending upon the products and the configurations involved; machine cycles, main storage, keyboard, display terminals, and so on. Some of the resources are local to the application program and some are remote. One APPC application program will request some resources from another system via the APPC application program running on that system.

As shipped, OS/400 provides several APPC applications. Display Station Pass-Through (DSPT), for example, allows a user on a local system to use the resources of a remote system as if locally attached.

Since APPC is part of SNA, it does not *enforce* security. As implemented on the AS/400, it provides a rich set of additional security facilities not available for other LU types. Users can also write their own APPC applications. This is covered in 11.15, “User-Written Applications and File Transfer Support” on page 11-28.

Advanced Peer-to-Peer Networking (APPN): APPN is an extension to the architectural definitions for PU type 2.1. It allows a point-to-point application link between systems that are not physically adjacent in the network. APPN takes advantage of the common transport network between the AS/400s (path control and data link control layers of SNA). In PU 2.1, the transport layer is implemented to provide the logical point-to-point connection between the LUs on a one-hop basis. Figure 10-5 shows the difference between APPC applications in an APPN and non-APPN network. Without APPN, application A running on the London system could not make direct APPC requests to application D for use of resources on the system in Edinburgh. Rather, the Manchester system would need to run an application that could communicate with both the London and Edinburgh systems. With APPN configured, application A can make a direct request to the application in Edinburgh.

The system in Manchester must be configured as an APPN networking (intermediate) node, and as such it only routes the data of the APPC session between the applications in Edinburgh and London.

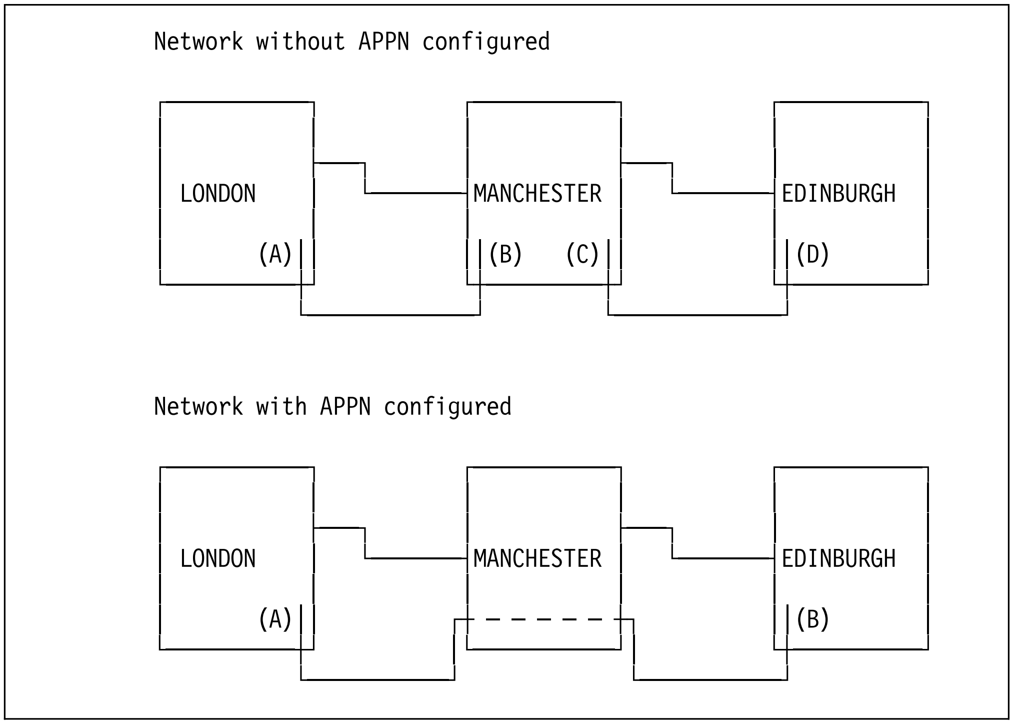


Figure 10-5. Distinction Between LU 6.2 Sessions in APPN and Non-APPN Networks. Without APPN, the only way for London to establish an APPC conversation with Edinburgh is via an intermediate APPC application (another LU-LU pair) in Manchester. With APPN, Manchester simply provides transport services for the LU-LU session between London and Edinburgh. Another APPC application is not required in Manchester.

AS/400 APPC applications can make extensive use of a network of systems configured for APPN. This is true for both user-written APPC programs and the ones supplied as part of OS/400, for example, DSPT, or DDM. Although we do not consider APPN by itself, it must be remembered that APPC applications, requesting some target resources, may actually be running on a system connected via many other intermediate systems (that is, not physically adjacent). The same APPC security can be applied regardless of the physical location of the source and target systems.

10.3.1 APPC Security

APPC supports a sophisticated set of security checks to validate not only the user to the system, but the location attempting to connect for a given type of work. It is possible to control the access to the target AS/400 by selectively allowing or denying access to certain APPC applications.

Specific APPC applications have different security implementations.

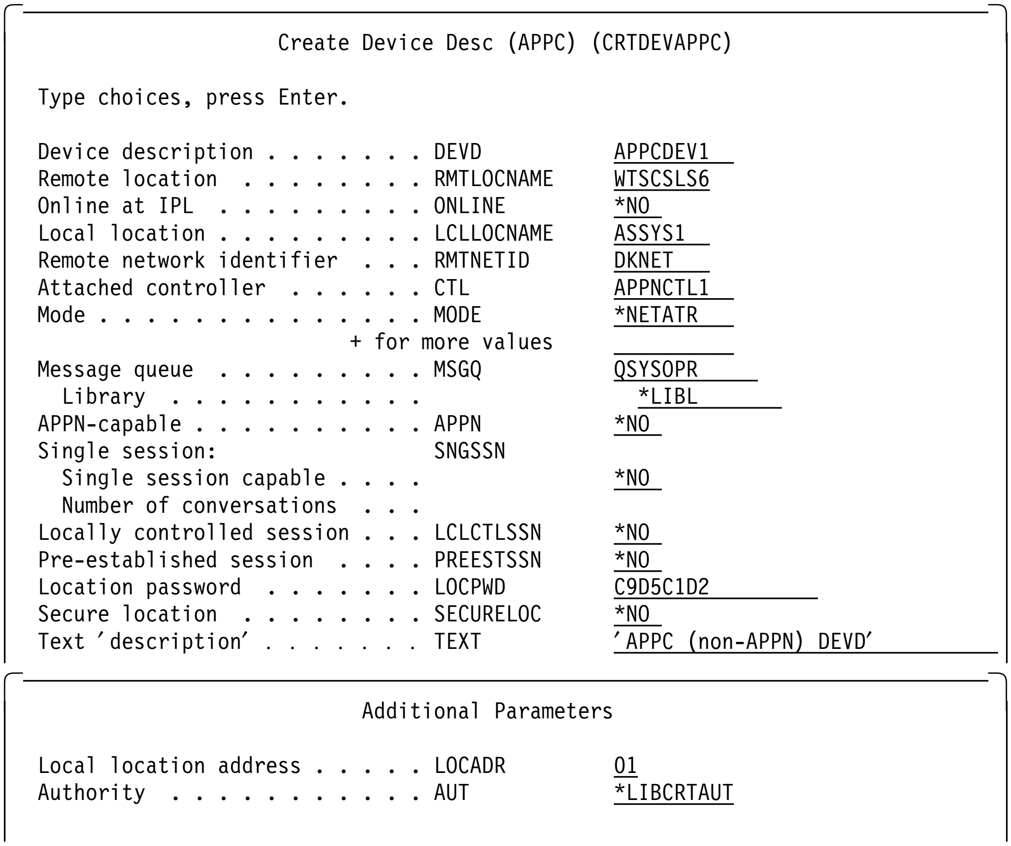
In addition to base level SNA security (that is, physical security, AS/400 resource security, and user profile security), APPC communications security consists of *Session Level Security* and *Conversation Level Security*.

10.3.2 Session Level Security

Setting up a session is analogous to establishing a telephone connection between two offices. In an APPC environment, no single system is responsible for the control of all sessions. The sessions your system is responsible for are called *local sessions*. The sessions the remote system is responsible for are called *remotely-controlled* sessions.

Session Level Security in APPC

This is established when the session is started (LU 6.2 Bind), and verifies the identity of the LU on both ends of the session. Session level security is achieved by using the location password (LOCPWD) parameter on the APPC device. Both sites must agree exactly. Their values can be *NONE. The value must be *NONE if the other system does not support session level security. Examples of these parameters are given in Figure 10-7 and Figure 10-6.



```
                       Create Device Desc (APPC) (CRTDEVAPPC)

 Type choices, press Enter.

 Device description . . . . . . . DEVD           APPCDEV1
 Remote location  . . . . . . . . RMTLOCNAME     WTSCSLS6
 Online at IPL  . . . . . . . . . ONLINE         *NO
 Local location . . . . . . . . . LCLLOCNAME     ASSYS1
 Remote network identifier  . . . RMTNETID       DKNET
 Attached controller  . . . . . . CTL            APPNCTL1
 Mode . . . . . . . . . . . . . . MODE           *NETATR
                        + for more values
 Message queue  . . . . . . . . . MSGQ           QSYSOPR
   Library  . . . . . . . . . . .                  *LIBL
 APPN-capable . . . . . . . . . . APPN           *NO
 Single session:                  SNGSSN
   Single session capable . . . .                *NO
   Number of conversations  . . .
 Locally controlled session . . . LCLCTLSSN      *NO
 Pre-established session  . . . . PREESTSSN      *NO
 Location password  . . . . . . . LOCPWD         C9D5C1D2
 Secure location  . . . . . . . . SECURELOC      *NO
 Text 'description' . . . . . . . TEXT           'APPC (non-APPN) DEVD'

                              Additional Parameters

 Local location address . . . . . LOCADR         01
 Authority  . . . . . . . . . . . AUT            *LIBCRTAUT
```

Figure 10-6. Location Password (Bind Validation) on the APPC Device Description. This panel is obtained using the CRTDEVAPPC command and pressing F9 for all parameters.

The CRTDEVAPPC command shows where the password used for bind validity checking is specified (LOCPWD). The password is used when a session is being established between the local location and the remote location identified in this command. If LOCPWD is *NONE (the default), no password is allowed to establish the session. The location password can only be seen when creating the device, after which it cannot be viewed. It must be between 2 and 16 hexadecimal characters.

Note: Both systems must use the same value. This value can be *NONE, which is a special value meaning no password is sent, and any password received is rejected.

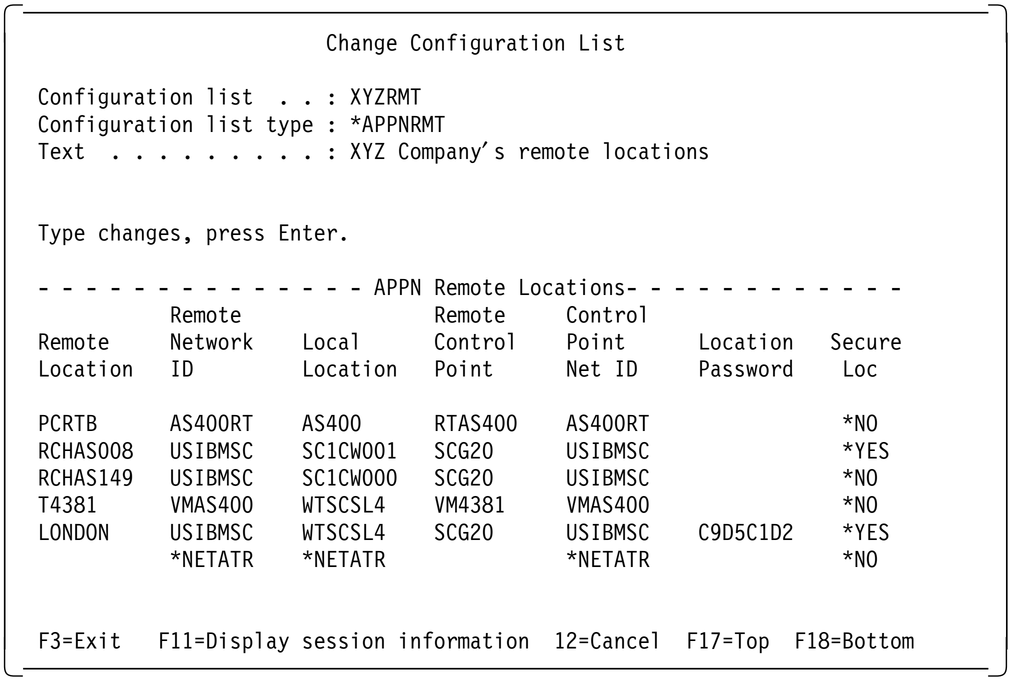
Using a location password enhances security considerably because of certainty about the identity of the remote system. This is especially important in X.25 networks and switched communications. More information on location passwords can be found in the *AS/400 Communications: APPN Networking Guide* or in the *AS/400 Communications: APPC Programmers' Guide*.

Session Level Security in APPN

Session level security is achieved by using the location password (LOCPWD) parameter in the APPN remote configuration list. Both sites must agree exactly.

The password is not checked in an intermediate node in an APPN network. The remote location configuration lists in the source and target systems must be used to establish a secure session.

The remote location configuration list and the APPC devices should be verified in a periodic audit.



```
                          Change Configuration List

 Configuration list  . . : XYZRMT
 Configuration list type : *APPNRMT
 Text  . . . . . . . . . : XYZ Company's remote locations


 Type changes, press Enter.

 - - - - - - - - - - - - - - APPN Remote Locations- - - - - - - - - - - -
            Remote                 Remote     Control
 Remote     Network    Local      Control    Point      Location   Secure
 Location   ID         Location   Point      Net ID     Password    Loc

 PCRTB      AS400RT    AS400      RTAS400    AS400RT                *NO
 RCHAS008   USIBMSC    SC1CW001   SCG20      USIBMSC                *YES
 RCHAS149   USIBMSC    SC1CW000   SCG20      USIBMSC                *NO
 T4381      VMAS400    WTSCSL4    VM4381     VMAS400                *NO
 LONDON     USIBMSC    WTSCSL4    SCG20      USIBMSC    C9D5C1D2    *YES
            *NETATR    *NETATR               *NETATR                *NO


 F3=Exit   F11=Display session information  12=Cancel  F17=Top  F18=Bottom
```

Figure 10-7. Location Password (Bind Validation) on the APPN Remote Location Configuration List. This panel is obtained using the WRKCFGL command and selecting option 2, change the list entries.

The APPN remote location configuration list shows where the password used for bind validity checking is specified. The password is used when a session is being established between the local location and the remote location identified in this list entry. If this field is blank, no password is required to establish the session. A location password can be entered or changed on this display. It must be between 2 and 16 hexadecimal characters (that is, A-F and 0-9).

Note: You can only see the password when you enter or change it. You are not able to display it.

Bind Validation: Table 10-5 shows the session level security which results from the combinations of location passwords between communicating AS/400s. When a bind fails no APPC communications will be possible between the two systems. When a bind is unsecure it indicates that security information will not be sent when conversations are established. A secure bind indicates that identities are established and conversation security will be supported. The location password is not used for unsecure binds.

Note: A secure bind with no password is supported.

	AS/400 QSecurity=10	AS/400 QSecurity>10 location password *NONE	AS/400 QSecurity>10 with location password
QSecurity=10	Bind is Unsecure	Bind is Unsecure	Bind Fails
QSecurity>10 location password *NONE	Bind is unsecure	Bind is secure	Bind fails
QSecurity>10 with location password	Bind fails	Bind fails	Bind is secure if passwords match

Table 10-5. Bind Validation Between Communicating AS/400s. The password (if used) is defined in the APPN remote location list or on the APPC device description and can be between 2 to 16 hexadecimal characters (0-9 and A-F).

10.3.3 Conversation Level Security

If setting up a session is analogous to establishing a telephone connection between two offices, the conversation is analogous to the dialog two persons exchange over the connection. Before the dialog can start the security is checked.

Password Protection

In V3R1 APPC substitutes another character string, called a *protected password*, for a user password when it starts a conversation. The password is no longer sent in the clear on LANs and communication lines. This is a standard implemented for *all* APPC connections in V3R1.

When both sides support it, such as DSPT between two AS/400s or Client Access/400 using V3R1, a protected password is used. If one side lacks the support the password is sent in the clear as in earlier releases of OS/400.

The user cannot change this support (turn it on or off).

The Conversation

The source side of a conversation:

- The application is responsible for providing adequate security information
- Starts the conversation by naming the remote program, and (optionally) specifying security information needed to access the remote program

The target side of a conversation:

- The system is responsible to verify adequate security
- Receives Program Start Request
- Verifies that security is adequate, program exists, and so on.
- Either passes the request to a pre-start job, or starts a new job to run the program

Conversation Level Security is done *after* a session has been acquired (part of which is passing the session level security checks). The source side of the conversation chooses the level of security to send. The target system chooses how much security is required to *successfully* start the conversation. This involves the SECURELOC parameter in the APPC device, or in the APPN remote location configuration list, and the DFTUSR parameter on the ADDCMNE command. Refer to "Subsystems" on page 10-10 about DFTUSR.

If the bind is secure (that is if both systems have QSECURITY > 10 and the location passwords match), a location can specify the amount of security information required from the other location for each remotely initiated APPC conversation. Conversations using SECURITY(SAME) or SECURITY(NONE) removes the need for applications to send passwords across the network together with user profiles. It simplifies network administration since passwords do not need to be the same on all systems, or prompted for in each APPC application.

Locations are either called *Secure Locations* or default to *Non-Secure Locations*. The parameter (*YES or *NO) for secure location is specified on the SECURELOC keyword in the CRTDEVAPPC command for APPC, and in the remote location configuration list for APPN.

A *Secure Location* represents acknowledgement by the target system that the security facilities of the source system are acceptable. The source location need only send the AVI (Already Verified Indicator) with the user profile. The target system will trust the security measures provided by the source system.

A *Non-Secure Location* indicates that the target system wants the source system either to send both a user profile and a password, or send no security information. In the latter case, when no security information is passed, the target system must have defined a default user profile in the communication entry in the subsystem used. This should be a user profile with authorities sufficient only to perform the required functions.

On an AS/400 the default user profile is added to the subsystem owning the communications device with the ADDCMNE or CHGCMNE commands before the subsystem is activated. This command specifies the devices from which EVOKEs are accepted. One communication entry may be set to apply to all APPC devices.

There are three levels of conversation security:

- SECURITY(SAME)
  - If the target system trusts the source system, send user ID with Already Verified Indicator (AVI)
  - If the target system does not trust the source system, send no password or user ID (see note)
- SECURITY(NONE) - send no password or user ID
- SECURITY(PGM) - send user ID, password, and/or profile specified by the application program

Note: In V3R1, if both sides support password protection, APPC allows you to use Enhanced SECURITY(SAME). That is, if the target system specifies SECURELOC(*NO) and the source system specifies SECURELOC(*SAME), the source system will retrieve the signed on user′s encrypted password and generate a protected password. This, of course, requires that the user ID and the password are the same on both systems.

When a program on a remote system wants to start a program on the local AS/400 it establishes *application program security* by sending security information with either the evoke command for ICF or the CMALLC call for CPI-C.

Secure locations have two aspects:

- It allows SECURITY(SAME) to send just a user ID with an Already Verified Indicator to this system, but doesn′t prevent SECURITY(PGM) to send user ID and password. SECURITY(SAME) prohibits passwords from being stolen by eavesdropping the line - because no password is being sent by the source system.
- If the source system is *not* secured properly, it exposes the target system as well.

The *APPC Programmer′s Guide* have more information about security in an APPC environment.

Chapter 11. Security in IBM Communications Applications

This chapter covers implementing security for an AS/400 in a *communications environment* using IBM′s communications applications. By communications environment we mean an AS/400 that also has users connected through other devices or attachments than the local 5250 or ASCII workstation controllers. Some examples are:

- Users connected by communications equipment like the 5X94 or 3X74 remote controllers
- Users connected to another computer system like an AS/400, ES/9000*, or RISC System/6000, which in turn is connected to the local AS/400
- Users or systems accessing the AS/400 on a token ring or Ethernet
- PC Support Users, which is covered in Chapter 12, "PC Support" on page 12-1

In theory, *all* AS/400 systems are part of a communications environment if they are using Electronic Customer Support (ECS). Security aspects of ECS are also covered in this section.

Connections over communications links is desirable for many reasons, such as to use resources (applications, databases, or hardware) not available on the user′s local system. An otherwise secure AS/400 is potentially compromised by the attachment of remote systems and users. In this case, the AS/400 is known as the target system.

Your AS/400 system may be connected to other systems with inadequate security measures in place. You may have no control over these systems but you will want to exclude possible security violations on your system due to poor security implementation on the connected systems. In this chapter we will concentrate on the AS/400 as the target for possible security violations.

Ensuring that facilities for remote users are available in a secure manner is a challenge. Security for AS/400s in a communications environment is achieved by a combination of:

- System values related to security
- AS/400 resource security
- Communications architecture
- AS/400 work management facilities

The system values related to security is covered in the section Chapter 2, "System Values and Network Attributes" on page 2-1 together with matching recommendations.

AS/400 resource security applies equally to stand-alone and communicating AS/400s, and you should read Chapter 4, "Resource Protection" on page 4-1 in conjunction with this section.

For information about security in the AS/400 communications configuration, SNA, APPC, and APPN see Chapter 10, "Communications Security in SNA" on page 10-1.

To learn more about managing security in a network environment, refer to the *Centralized Security Administration In An AS/400 Peer Network*, GG24-3719.

A discussion of the Trusted Network Interpretation as issued by the National Computer Security Center which describes network system security guidelines can be found in Appendix J, “Trusted Network Interpretation (TNI)” on page J-1.

11.1 Security in IBM Supplied Communications Applications

The following sections detail some of the considerations for the following IBM provided communications applications:

- Host Command Facility/Distributed Host Command Facility (HCF/DHCF)
- Network Routing Facility (NRF)
- SNA Primary LU2 Support (SPLS)
- Distributed Data Management (DDM)
- Distributed Relational Database Architecture (DRDA)
- Display Station Passthrough (DSPT)
- Netview Distribution Manager (NDM) and Distributed Systems Networking Executive (DSNX)
- SNA Distribution Services (SNADS)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- TCP/IP File Server Support/400 (FSS)
- 3270 Device Emulation
- Remote Job Entry (RJE)
- Electronic Customer Support (ECS)
- Open Systems Interconnection (OSI)

Section 11.15, “User-Written Applications and File Transfer Support” on page 11-28 considers user-written application programs together with the IBM supplied File Transfer Support subroutines that can be used in such applications.

Note: If you are auditing communications security you should refer to the relevant sections in this chapter for more information about the communications products that may be used in your installation.

A particular security concern is the user profile on the target system used by the application. A user may be authorized to many commands and objects on their own system but need not have as many capabilities on the target system. It is important that the security implementation provides sufficient reporting capabilities, should a potential exposure need to be investigated.

11.2 Distributed Host Command Facility (DHCF)

The Distributed Host Command Facility of OS/400 allows the users of display stations of an ES/9000*, using the Host Command Facility (HCF) companion program under ACF/VTAM*, to sign on to an AS/400 as a remote AS/400 work station.

Typical use for HCF is for Network Management, for example, the Help Desk.

DHCF is an LU 0 application and works like remote 3270 attached terminals. As such the only security imposed is user identification, a valid user profile and password must be used to sign on to the AS/400.

This raises several security considerations:

- HCF users need to be issued user profiles with sufficient authority only to do the required work.
- Users need user profiles and passwords to access *all* the AS/400s in the network, in order to provide the support required.
- HCF on the host system can only contact adjacently configured AS/400s. To access non-adjacent AS/400s, display station pass-through must be used. (See 11.7, "Display Station Pass-Through (DSPT)" on page 11-9 for further details).
- Issuing the same user profile and password for all systems to be used only by the HCF user, means the user need only be issued (and remember) a single combination. However, the exposure is that all systems in the network are vulnerable if this gets into the ″wrong″ hands.
- The HCF user profile, on a given AS/400, is infrequently and randomly used. This emphasizes the importance of good network security management. The security practices implemented on one AS/400 in the network should apply equally across all the AS/400s in the entire network.

11.2.1 Recommendations for HCF/DHCF

- Issue *one* user profile and password for each HCF user for all AS/400s in the network. This is to be provided solely for the purposes of the HCF users.
- At suitable time intervals change all the HCF passwords simultaneously.
- Journal the HCF user profiles. The security administrator for each AS/400 should be responsible for monitoring the HCF user profile activities on a regular basis. An example of journaling user profiles is given in Chapter 6, "Audit Journal" on page 6-1.

Note: Remember the exposure: if this user profile and password gets into the ″wrong″ hands, it could be used to access *all* the AS/400s in the network. However, the alternative is for the HCF user to remember user profiles and passwords for many AS/400s in the network, together with the password maintenance implications. In such a situation, it is more likely that the HCF user would need to write down the user profiles and passwords, which is a far greater exposure.

Also note that passwords are not encrypted by DHCF on the communications lines, they are transmitted in the clear.

11.3 Network Routing Facility (NRF)

Network Routing Facility (NRF) is an alternative to Host Command Facility (HCF) allowing mainframe attached 3270 devices to access AS/400 functions.

Integrated NRF support in OS/400 allows the 3270 device to pass through the 3745 controller directly to the AS/400. This avoids the use of host (ES/9000) resources and provides increased connect speed. NRF also supports host (LU 1) SNA Character Set (SCS) printers. It means that the host users can get the printed output from the AS/400 applications routed back to their host SCS printers attached to a 3x74 control unit. The NRF licensed program operates on the 3745 communications controller.

11.3.1 Recommendations for NRF

This support has the same security requirements as HCF/DHCF and other remote 3270 attachments. Remote interactive 3270 display activity will always require a valid AS/400 user profile and password to sign on. From a security viewpoint NRF should be considered equally to attachment of remote 5250 devices.

11.4 SNA Primary LU2 (3270 Type Terminal) Support (SPLS)

SNA Primary LU2 Support (SPLS) improves the AS/400 participation in an SNA backbone network and can been seen as an alternative to HCF/DHCF and NRF.

SPLS enables any 3270 (LU 2) terminal user in the network to access any AS/400 system or any ES/9000 system in the network with the same user and network interface.

At least one host using VTAM V3R4 and NCP V5R4 is required for OS/400 SPLS. Like NRF, SPLS has support for host attached SCS printers.

11.4.1 Recommendations for SPLS

This support has the same security requirements as HCF/DHCF, NRF, and other remote 3270 attachments. Remote interactive 3270 display activity will always require a valid AS/400 user profile and password to sign on. From a security viewpoint SPLS should be considered equally to attachment of remote 5250 devices.

11.5 Distributed Data Management (DDM)

DDM is part of the OS/400 operating system. It provides the capability for application programs to access data files that reside on remote systems that support DDM. DDM also allows remote systems to access data files on the local AS/400. DDM provides the facility to use the Submit Remote Command (SBMRMTCMD) command. This enables a user at a source system to submit CL commands to a target system, without the need for interactive signon. Systems using DDM communicate using LU 6.2 (APPC).

Using DDM, an application can retrieve, add, change, and delete data records that exist in the target system. It also allows for file-related operations such as creating, deleting, renaming, or copying a file to or from the target system.

The DDM job runs under a user profile on the target system. The authority of this user profile governs the access of the DDM job.

DDM requires that a device file (DDM file) is created on the source system. This is not a data file, but rather appears to the source application as a local database file. The DDM file includes the name of the remote file and library, a remote location name, and device description name, to route the DDM request to the target system where the database file actually exists.

Access to target AS/400 data can be limited by using the standard authority to files, and by using an optional user exit program in the DDM environment at the target system (specified in the network attributes).

Security checking is performed when a remote user accesses an AS/400 file. The remote user must be authorized to perform the operation, or the DDM request is rejected. When DDM is used, the data resources of each system in the DDM environment can be protected by normal session level and location security, with the following differences:

- SECURELOC(*YES). DDM will *always* use the Already Verified Indicator (AVI) and will never request the sending of a user password with the user profile.
- SECURELOC(*NO). DDM will never send a user profile and password for non-secure locations. A *default user profile* must be defined in the communication entry in the subsystem used for DDM.

See 10.3.1, “APPC Security” on page 10-13 for more information about how to set SECURELOC.

DDMACC (DDM access), a value set in the system network attributes, controls the DDM access allowed. Use DSPNETA to display the value and CHGNETA to change it.

11.5.1 DDM Location Security

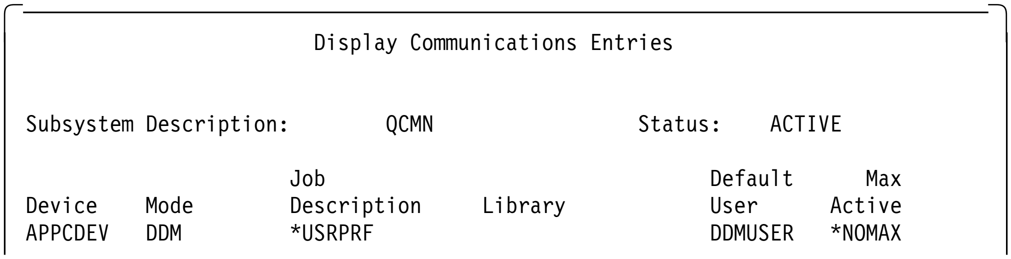
Care must be taken when creating the user profiles on the target system to be used for DDM. For ease of security management, it may be desirable to duplicate all user profiles across all systems in the network. However, it may not be necessary (or even desirable) to have a user profile on the target system with the same rights as are available to the user on the source system. For example, a user with *ALLOBJ authority on the source system may only need to have access as *USER on the target (DDM) system.

If SECURELOC(*NO) is specified, the target system allows the source system requests to be handled using the default user profile defined in the communications entry of the subsystem used for DDM. This means that all DDM users corresponding to a given default will have exactly the same capabilities. **This is not usually an acceptable security implementation.**

A secure location (SECURELOC(*YES)) definition provides far more effective security, a means to determine the identity of the DDM user, but at the cost of maintaining more user profiles on the target DDM system. In this case you will add your own communications entries to the subsystem used for DDM.

By using the communications subsystem (QCMN), or one you create, and making specific communications entries, the target system can be more certain of the location requesting DDM access. The default user name should reflect the nature of the application being utilized (for example DDMUSER). An example of specifying subsystem communications entries is given in Figure 11-1 on page 11-6.

You add communications entries to the subsystem with the Add Communications Entry (ADDCMNE) command.



```
                           Display Communications Entries


 Subsystem Description:        QCMN               Status:    ACTIVE

                     Job                                  Default      Max
 Device    Mode      Description     Library              User      Active
 APPCDEV   DDM       *USRPRF                              DDMUSER   *NOMAX
```

*Figure 11-1. Specific Communications Subsystem Entry in QCMN. DDM requests from device APPCDEV will be accepted through this communications entry in the communication subsystem QCMN. If no user profile is supplied, the requester will access the target system with the user profile DDMUSER. This user profile has rights sufficient only to access the required files. It has a password set to *NONE to prevent interactive signon.*

Note: The default user is used for *all* remote requests (not only DDM) that come from communication devices that match this communication entry, such as Display Station Pass-Through and SNADS. This could be a possible security breach.

11.5.2 DDM Conversations

An important security consideration relates to how DDM conversations are ended. For DDM, there are at least two jobs started - one on the target system and one on the source system. Many applications could be running in one source job. For each source DDM job, there is a separate DDM conversation with the target DDM job in the remote location. If many DDM files within the same source job specify the same remote location, they will share the same conversation. When the source DDM job closes the DDM file, the DDM conversation and the target DDM job ends, unless:

- The value of the DDMCNV attribute in the CHGJOB command specifies *KEEP (the default). The DDM conversation remains active and waits for another DDM request to be started.
- File locks established during the job still exist.

*KEEP provides better performance than constantly ending and re-starting conversations. However, security information has already been passed to establish the conversation. It may not be desirable to maintain certain conversations between systems, for example, to access the payroll file held at the target system. In such cases, the source and target system′s security should, if possible, be implemented cooperatively. Sensitive DDM applications on the source system should run in a strictly controlled job environment (CRTJOBD) that is not available to other applications. Set public authority to *EXCLUDE for the job description and use an authorization list to specifically grant authorities to the users.

The authority to the DDM file at the source system is outside the control of the target system. Authority to the data files at the target system can be strictly controlled using normal file authorities. In addition, the target system should make use of the DDMACC parameter on the network attributes. This is covered in 11.5.4, “DDM Access Parameter (DDMACC)” on page 11-7.

11.5.3 Submit Remote Command

DDM provides the capability for a user on a source system to send a CL command to execute on the target system. The DDM file on the source system is used to identify the remote location (target system) to which the command is sent. The primary purpose of the command is for file management and authorization related to the DDM application. However, many commands can be issued that need not be related to such activities. Commands can be issued in this environment as long as they do not attempt to display output to the workstation. A complete list of commands that can be issued using SBMRMTCMD is given in the *AS/400 Distributed Data Base Guide*. The user must be authorized both to the CL command and the objects that the command will operate upon. The authority to the SBMRMTCMD command on the source system should be set to PUBLIC *EXCLUDE. The GRTOBJAUT command can then be used to grant users authority to the SBMRMTCMD command.

11.5.4 DDM Access Parameter (DDMACC)

The DDMACC parameter in the network attributes is used to indicate whether an AS/400 will accept DDM requests from other systems. The value can be changed using the CHGNETA command, or viewed using the DSPNETA command. The possible values are shown in Table 11-1.

Value	*Meaning*
*REJECT	No DDM requests are allowed from remote systems. However, this system can still use DDM to access other systems.
*OBJAUT	DDM requests are allowed from remote systems. Normal AS/400 object authority applies. The authorizations are dependent upon the user profiles specified (for SECURELOC(*YES)) or on the default user profile authorities on the target system.
qual-pgm-name	The name of a user exit program and its library used to *supplement* normal object level authority.

Table 11-1. Possible Values for DDMACC on the Network Attributes

Refer to the *AS/400 Security Reference* for more information.

DDM requests can be rejected or accepted subject to normal object authority implemented on the target system. We recommend that DDMACC is immediately changed to *REJECT if DDM access (including the use of SBMRMTCMD or PC Support functions) is not required.

The option of specifying a user-written exit program can be used. This controls whether a given user of a specific source system can use a particular DDM command to access a specific file on the target system.

Examples of user exit programs for DDMACC are shown in Appendix D, “Program Used with DDMACC on Network Attributes” on page D-1 and Figure 12-3 on page 12-16 (to reject the SBMRMTCMD command).

11.5.5 Recommendations for DDM

The following are recommendations for implementing good security for systems that use DDM functions:

1. If no DDM facilities are to be provided at the target system, change the DDMACC parameter of the Network Attributes to *REJECT.

   Note: PC Support/400 also uses the DDMACC parameter. For more information about this subject see Chapter 12, "PC Support" on page 12-1.

2. Where DDM facilities are to be provided, and object level security is not considered sufficient control for DDM access, specify a user exit program on the DDMACC parameter. Reject all DDM requests except from specified remote locations, and from unauthorized users. You may want to use this facility only if special safeguards are needed.
3. If some DDM functions are required, but no remote functions are to be performed (for example no remote commands), include in the DDMACC program a routine to reject the SBMRMTCMD command.
4. Make communications entries in the subsystem used for DDM requests for specific remote locations or devices.
5. Create a default user profile (DDMUSER), to be used when SECURELOC(*NO) applies, with rights sufficient only for the required functions, and change the default user in the communications subsystem used for DDM.
6. Use standard object authorization for the files to be accessed.
7. If cooperation between the target and source systems for security implementation is feasible, the source system should:
   - Create the DDM files with *PUBLIC *EXCLUDE and grant authority only to users who need DDM access to the target system.
   - Ensure that the MODD for the DDM conversation has maximum sessions and conversations set to the minimum required for the DDM conversations.
   - For sensitive applications, the value of the DDMCNV attribute for the source job should be changed to *DROP (CHGJOB command). This ensures that when the conversation has ended it is dropped.
   - Change the authority of the SBMRMTCMD command (on the source and the target system) to *PUBLIC *EXCLUDE and grant specific authority only to those who need it.
8. In V3R1 Enhanced Security Same can be used:
   - Set SECURELOC(*NO).
   - DDM users must have the same user ID and password on both systems.
   - APPC will send protected passwords for these users.

11.6 Distributed Relational Database Architecture (DRDA)

The *AS/400 Distributed Relational Database Guide* describes many security implications for a network of relational databases. The key security issue will probably be whether a unique user profile must be created for each remote SQL user on each system in the network. OS/400, DB2, and SQL/DS support default or generic user profiles (with different levels of authority) so that a group of remote users can use one user profile, which reduces the workload for security administrators. However, it significantly reduces auditability and granularity of authorizations, so in many customer environments it will not be acceptable.

DRDA has defined that each system which supports DRDA, supports SNA security levels of SAME and PGM on both the Application Requester (AR) and the Application Server (AS). Security SAME involves only sending a user profile to the AS, and security PGM involves sending a user profile and a password. The AS/400 supports both security levels as an AS and AR.

If default or generic user profiles are not used on the application requester when connecting an application server, the user profile name specified in the USER parameter of the CONNECT statement, or the user profile name of the person starting the job (if USER was not specified), will be sent to the application server with an indication that it has been verified by the application requester. This indication is implemented by defining the remote location as a secure location either in its device description or in the remote configuration list if APPN is being used. The requester′s user profile must be defined on the server system. It should only be given the authority needed to get the job done. This security method should encourage you to create unique user profiles across all systems. Otherwise a user profile called ″ADMIN″ on system A might find they could drop (delete) critical tables on system B if there was a user called ″ADMIN″ on that system with suitable authorizations. See 10.3.1, “APPC Security” on page 10-13 about how to specify a secure location.

The DDMACC (DDM access parameter) network attribute (explained in 11.5.4, “DDM Access Parameter (DDMACC)” on page 11-7) applies *only* to DDM file operations and cannot be used to control or limit DRDA accesses.

11.6.1 Recommendations for DRDA

As the previous section explained, you must decided if you want to create unique user profiles for each remote SQL user on each system in the network or use the default/generic user profile approach.

This first option gives you better control and auditability, but it increases user profile management. The latter approach reduces user profile management but does not provide the same granularity in granting authorizations. Your requirements for security and auditability will govern which option you choose.

11.7 Display Station Pass-Through (DSPT)

DSPT allows remotely or locally attached source system users to interactively signon to another IBM S/36, S/38, or AS/400 in a network. DSPT is an LU 6.2 based application. By making use of a network configured for APPN, users can pass-through to any appropriately configured system in the network by the use of a single command.

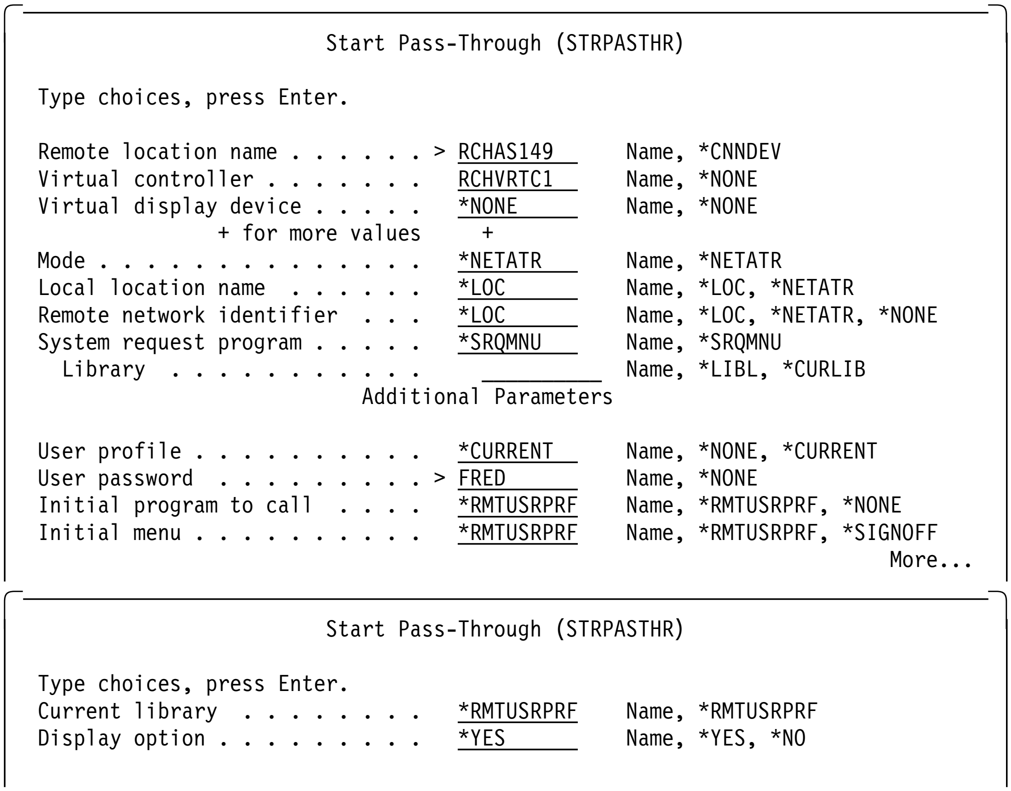
DSPT implements LU 6.2 based security with the following additional considerations:

- User identification
- Configuration of virtual controllers descriptions (VRTCTLD) and virtual device descriptions (VRTDEVD)
- System value QRMTSIGN

11.7.1 DSPT User Identification

A user on a source system starts a DSPT session to a target system by issuing the STRPASTHR command. The target system is selected by the remote location name on the command. There are two different ways a user can sign on, depending upon the configuration of the target system. In the first method, after issuing the command, the user is presented with the target system signon screen. A user profile and password must be entered and normal security validation is performed. Alternatively, the user can select automatic signon to the target system. Automatic signon is achieved when the user provides, on the STRPASTHR command, either a different user profile and password, or uses the same user profile and password as used on the source system (*CURRENT). The signon display on the target system will not be displayed if the profile and password are valid.

An example of the STRPASTHR command is shown in Figure 11-2.



```
                        Start Pass-Through (STRPASTHR)

 Type choices, press Enter.

 Remote location name . . . . . . > RCHAS149      Name, *CNNDEV
 Virtual controller . . . . . . .   RCHVRTC1      Name, *NONE
 Virtual display device . . . . .   *NONE         Name, *NONE
               + for more values      +
 Mode . . . . . . . . . . . . . .   *NETATR       Name, *NETATR
 Local location name  . . . . . .   *LOC          Name, *LOC, *NETATR
 Remote network identifier  . . .   *LOC          Name, *LOC, *NETATR, *NONE
 System request program . . . . .   *SRQMNU       Name, *SRQMNU
   Library  . . . . . . . . . . .     __________  Name, *LIBL, *CURLIB
                           Additional Parameters

 User profile . . . . . . . . . .   *CURRENT      Name, *NONE, *CURRENT
 User password  . . . . . . . . . > FRED          Name, *NONE
 Initial program to call  . . . .   *RMTUSRPRF    Name, *RMTUSRPRF, *NONE
 Initial menu . . . . . . . . . .   *RMTUSRPRF    Name, *RMTUSRPRF, *SIGNOFF
                                                                      More...
```

```
                        Start Pass-Through (STRPASTHR)

 Type choices, press Enter.
 Current library  . . . . . . . .   *RMTUSRPRF    Name, *RMTUSRPRF
 Display option . . . . . . . . .   *YES          Name, *YES, *NO
```

*Figure 11-2. Start Display Station Pass-Through Command (STRPASTHR). The source system user has specified the name of the remote location and a virtual controller to use for the DSPT session. In addition, the user is requesting automatic signon at the target system by entering the user profile and password in the additional parameters section. In this case the same user profile on both source and target systems (*CURRENT) is used.*

11.7.2 DSPT Virtual Configuration Descriptions

In order to use DSPT, there must be controller and device descriptions on the target system that represents the source system. They are called *Virtual Controller* and *Virtual Device* descriptions.

These virtual descriptions can be configured manually using the CRTCTLVWS and CRTDEVDSP (type *VRT) commands, or created automatically using the QAUTOVRT system value.

If QAUTOVRT is greater than 0 a virtual controller, QPACTL00, is automatically created if it does not exist, and the devices that are automatically created are linked to this controller. A virtual controller can have a maximum of 250 virtual devices linked to it. When that limit is reached a new virtual controller, QPACTL01, is created, and so on. Virtual devices are owned by the user profile on the remote system that triggered the creation.

QAUTOVRT is valid for DSPT and TELNET. It has no influence on PC Support/400 Workstation Function.

If QMAXSGNACN is set to 1 it represents a security exposure that is not acceptable: If QAUTOVRT is set to other than 0, virtual controllers and devices will be automatically configured when the DSPT programs request them. Without auto-configuration, a user has a limited number of attempts to break in. This is determined by the values for QMAXSIGN and QMAXSGNACN. When the maximum number of signon attempts is exceeded, one of the following will happen:

- **QMAXSGNACN = 1** - the virtual device used is varied off. However, the user will be assigned another virtual device and can continue to try until all attempts have failed (QMAXSIGN is ″exhausted″). With auto-configuration this is not acceptable, since each time a virtual device is varied off, an new STRPASTHR command will cause another virtual device to be automatically created, up to the maximum set by QAUTOVRT. In this case, the number of attempts a user has could be very high.
- **QMAXSGNACN = 2** - the user profile is disabled, but the virtual device used is still varied on.
- **QMAXSGNACN = 3** - the virtual device used is varied off, and the user profile is disabled.

Recommendation

We strongly recommend that QMAXSGNACN is left at 3, the default value when the system is shipped.

To make sure that only known virtual configuration objects can be used for DSPT, we suggest the use of the following technique, when establishing the DSPT configurations:

1. Set the QAUTOVRT system value to the maximum value of 9,999 and let all possible remote DSPT users sign on at the same time. The system will create all virtual controller and device descriptions that might be needed.
2. Change QAUTOVRT to 0 (CHGSYSVAL QAUTOVRT 0). No more virtual descriptions will automatically be created, even if an incoming call requests it.
3. Since auto-configuration of virtual descriptions uses the same naming convention on all AS/400s, it presents a potential security exposure. Rename the virtual descriptions supplying a new name to the auto-configured objects. Only ″informed″ source systems can make use of these renamed virtual descriptions with their new user-supplied names.

4. Users on the source system now have to specify a ″real″ name of a virtual controller or device description when using the STRPASTHR command.

When a source system user does not specify a virtual description on the STRPASTHR command the source system uses a special VRTCTLD (QPACTL00) to tell the target system that auto-configuration is required (or must have been used). If that is not the case the user may still be able to pass-through. The target system operator need only create a VRTCTLD called QPACTL00 with an appropriate virtual device, and the user on the source system will be able to access the target system. To prevent the target operator configuring the special controller, we suggest that the CRTCTLVWS command be changed to *PUBLIC *EXCLUDE, and specific authorities granted only to those responsible for configuration.

If system value QLMTSECOFR is set to ′1′, users with the *ALLOBJ or *SERVICE authority will have to be explicitly authorized to use APPC devices as the system auto-configures them. Authorization can only be given for existing device definitions, so it cannot be used before the auto-configured devices have been created. This means that you may have to take special measures to give *ALLOBJ and *SERVICE special authorities to the user profile used for central network problem determination and system administration, (see 11.2, “Distributed Host Command Facility (DHCF)” on page 11-3).

11.7.3 DSPT System Value QRMTSIGN

Another parameter that governs DSPT eligibility is the system value QRMTSIGN. Valid values are shown in Table 11-2.

Value	Meaning
*REJECT	All pass-through operations to this target system are rejected. This is the best method to prevent DSPT access to the target system. However, the DSPT jobs do start briefly. To prevent *any* DSPT activity, then the communications entry in the subsystem should be changed.
*FRCSIGNON	Force signon (the default value). All pass-through sessions started for this system must go through normal signon procedures. If DSPT is used, this is possibly the safest implementation, since it forces the DSPT users to correctly identify themselves to the target AS/400.
*SAMEPRF	Same profile. The signon display at the target system will not be displayed if the user profile on both the source and target system are the same. Password verification is done before the target DSPT program is used. If you supply a different user profile on the STRPASTHR command the pass-through attempt will fail with a security error. For pass-through attempts not requesting automatic signon, the signon screen will be displayed.
*VERIFY	For source system attempts requesting automatic signon, the source system user bypasses the target signon screen. The user profile may be different from the source system user profile. The target user profile must exist and password validation will be performed.
″program″ and ″library″	The name of a program (and library containing it) that runs at the beginning and end of every DSPT job. Additional security checking can be implemented to limit the access by source system users.

Table 11-2. System Value QRMTSIGN

With SECURELOC(*YES), the target system ″trusts″ the security measures provided by the source system. Whereas this may be a satisfactory condition, it enables the situation where QRMTSIGN is either *SAMEPRF or *VERIFY and a user profile is the same on the source and target system, automatic signon will occur without supplying a valid password. This would allow the QSECOFR user profile, for example, to be used on the target system with equal rights as on the source system, which may not be desirable. Using a QRMTSIGN exit program enables an installation to restrict user access via DSPT to only authorized users at known locations. See 10.3.1, “APPC Security” on page 10-13 about how to set SECURELOC(*YES).

Appendix E, “Example Exit Program for QRMTSIGN System Value” on page E-1 shows an example of an exit program for QRMTSIGN. The program checks the remote location of the DSPT requests. Only requests from the given location (SECURELOC(*NO)) and by non-″Q″ user profiles are accepted. In this example, QSECOFR on the source system would not be able to sign on as QSECOFR on the target. However, you may have a single person assigned the QSECOFR user profile for several AS/400s, who relies on being able to use DSPT. In such a case it might be necessary to allow DSPT by the QSECOFR profile, but not any other ″Q″ user profiles. The program could be modified to accommodate this and many other situations.

More information on QRMTSIGN and exit program examples can be found in the *AS/400 Communications: Remote Work Station Guide*.

11.7.4 Recommendations for DSPT

The following are recommendations for implementing good security for DSPT on the target AS/400:

1. If no DSPT sessions are to be allowed, including the workstation function (WSF) of PC Support (see 12.4.1, “Work Station Function” on page 12-11 for more details about WSF), set the system value QRMTSIGN to *REJECT.
2. To control remote access by user profiles with *ALLOBJ or *SERVICE authority, set the system value QLMTSECOFR set to ″1″. Use the GRTOBJAUT command to specifically grant rights only to those user profiles requiring *ALLOBJ access from a source system.
3. Add communications entries to the subsystem used for DSPT for selected remote locations. Add a default user profile that has limited authority. The profile will be used when the source system does not supply one. The profile need only have the following authority:
    - To run the IBM supplied program QPAPAST2 in library QSYS (this program is called to initiate the DSPT application on the target system)
    - To the APPC devices used for DSPT
    - To the DSPT job description
    - To the DSPT VRTCTLDs and VRTDEVDs
    - Sufficient for the jobs to be run
4. To restrict the virtual configuration objects that DSPT will use, implement the suggestions in 11.7.2, “DSPT Virtual Configuration Descriptions” on page 11-10. Make specific workstation entries in the subsystem used for DSPT for the virtual devices used, and remove the default entries (if any).
5. For APPN(*YES), define location password (LOCPWD) entries (for bind validation) in the remote location configuration list.
6. If DSPT is allowed and the security of the source system is considered acceptable:
    - Specify SECURELOC(*YES) on the APPC devices or in the remote location configuration list.

   - If automatic signon is required from that location, set the system value QRMTSIGN to *SAMEPRF. This forces the user to have the same user profile on the source and the target systems and helps identify specific user activity. However, the user profile on the target system should *not* necessarily have the same rights as on the source system, only sufficient rights to do the required job. Although this is at the expense of additional user profile and password management, it reduces the risk of a user accessing the target system with a user profile not within the users rights, or gained illegally. (Beware of the QSECOFR implications).
   - If automatic signon is not required from the remote systems, set the system value QRMTSIGN to *FRCSIGNON.

7. If DSPT is allowed and where the security of the source system is *not* considered acceptable:
   - Specify SECURELOC(*NO).
   - Where automatic signon is not required, set the system value QRMTSIGN to *FRCSIGNON. This ensures that the user is presented with the target system signon screen and must enter a valid user profile and password for that system.
   - For additional security checking, specify the name of an exit program and library on the QRMTSIGN system value. Reject attempts to sign on with ′Qxxxx′ user profiles. On a given AS/400, there may be many remote locations defined for many APPC applications. Therefore, the program should also reject attempts to start DSPT sessions from all but specified remote locations.

8. Consider also the following actions:
   - Create a job description to be used for the DSPT session. Limit access to libraries other than the ones needed for the DSPT job.
   - Define *PUBLIC authority *EXCLUDE for the virtual devices created for the DSPT session to prevent non-DSPT users from using them. Define the authority for the virtual devices as *CHANGE for the valid DSPT users.
   - Specify an initial menu in the DSPT user profiles that limits their activities.
   - Create the user profile for the DSPT user with limited capability (LMTCPB(*YES)). Reference the job description created for the DSPT user and the menu created for the pass-through user.
   - Force the DSPT user to use the ENDPASTHR command when leaving the target system. SIGNOFF presents the target system signon screen and does not end the DSPT session. Exclude the user profile created for the DSPT session from the command SIGNOFF in order to prevent signing off without ending the DSPT session.

For additional information on Display Station Pass-Through including configuration and security see the *AS/400 Communications: Remote Workstation Guide*.

11.8 NetView/DM and Distributed Systems Networking Executive

NetView*/DM (NDM) on a host ES/9000 and Distributed Systems Node Executive (DSNX) on AS/400 and other distributed nodes are applications that allow change and distribution management functions from a host ES/9000. The ES/9000 can retrieve data from an AS/400 (such as a development system) and send to other AS/400s in a network. This data can be files, job streams, commands, software fixes, and so on. It also provides the capability to delete AS/400 objects.

Distribution can be to one or many nodes simultaneously and can be coordinated to occur at predetermined time intervals, usually when the AS/400 and network activity is lowest.

NDM and DSNX are LU 0 based applications. Sessions can only be initiated by the host.

11.8.1 Configuring NDM and DSNX

Implementation involves configuration at the host and AS/400. In NDM, the name of the AS/400 node must be defined and must match the system name in the network attributes on the AS/400. Dependent upon the capabilities of the host controller configured (CRTCTLHOST), there will be an exchange of IDs (SSCPID, RMTCPNAME and so on, - refer to Table 10-2 on page 10-7) at session establishment. The parameters defined at the host and on the AS/400 must match. In addition, an application ID must be defined at the host (in VTAM) that matches the APPID parameter on the AS/400 device description (created using the CRTDEVSNUF command).

A user profile and password must be supplied at the host specifically for the DSNX application. This must be a valid user profile on the AS/400. A default is supplied (QDSNX); another user profile could be defined and used for this purpose. The password is **not encrypted** and can be viewed by anyone having access to the interactive user interface of NetView/DM (GIX - Generalized Interactive Executive). The DSNX user profile must have the specific rights to perform the activities required by NetView/DM during a session. Ensuring the security of the user profiles and passwords held at the host is outside the control of the target AS/400 security. Consequently, other methods must be used to ensure the target AS/400 is not subject to unauthorized use.

By default, the subsystems QBASE and QCMN contain generic communications entries for the DSNX session. However, from a system management viewpoint, it is preferable to isolate the DSNX jobs and use the QDSNX subsystem. This also enhances security since communications entries should only be added for the specific named devices that will be used for the NDM-DSNX sessions. It is not recommended that generic (for example, DSNX*), global (*ALL), or type (for example, *SNUF) entries be made.

The user profile specified at the host is added as a default user profile to the QDSNX subsystem. Since this user profile and its password are held in the clear at the host, it is recommended that the password is changed to *NONE, so that no interactive sessions can be started. The IBM supplied user profile (QDSNX) already has the password set to *NONE. This does not, however, prevent the use of a DSNX user profile on a source system from being used to issue the submit remote command (SBMRMTCMD) to the target system. This has the following potential security exposure:

- Target system sets SECURELOC(*NO) for the source system. In this case, no user profile is sent to the target. The default user profile for DDM (SBMRMTCMD uses DDM functions) will be used for the command to be processed on the target system.
- Target system sets SECURELOC(*YES) for the source system. If all the AS/400s in the network use the same default user profile (for example, QDSNX) for the DSNX sessions, the user profile of the source system will be sent to the target and the command will be accepted.

See 10.3.1, “APPC Security” on page 10-13 about how to set SECURELOC.

To prevent such use, the network attribute, DDMACC should either be set to *REJECT (in which case SBMRMTCMD will be prevented for all users) or to use a DDMACC exit program. The program should reject all attempts by the DSNX user profile to use the SBMRMTCMD. Refer to the discussion in 11.5.4, “DDM Access Parameter (DDMACC)” on page 11-7 and the example program given in Appendix D, “Program Used with DDMACC on Network Attributes” on page D-1.

11.8.2 Recommendations for QDSNX

The following summarizes the recommendations for implementing security for AS/400s running DSNX:

1. Set the password of the QDSNX user profile set to *NONE to prevent a user (who may have obtained the user profile and password from GIX at the host) from signing on interactively. Alternatively, create a different user profile with password *NONE for the same function.
2. Do not use the subsystems QBASE or QCMN, rather use QDSNX or create a subsystem for this purpose. Make specific communications entries in the subsystem for the SNUF devices that are to be used for the DSNX sessions.
3. Change the Network Attribute DDMACC to *REJECT or use an exit program to reject use of the SBMRMTCMD by DSNX user profiles.

11.9 SNA Distribution Services (SNADS)

SNADS is an architectural extension to SNA that provides the asynchronous (delayed) delivery of data in a network. Delivery waits until the network connections are available. SNADS provides the capability for applications to distribute objects, such as documents, files, and messages to other systems that are directly or indirectly attached. SNADS works with other AS/400s as well as with S/36, S/38, ES/9000 (for PROFS), and for the TCP/IP mail functions of the AS/400.

SNADS also supports the use of the SBMNETJOB command. This allows a user on a source system to send a job stream to be executed on the target system. The SBMNETJOB command is discussed in 11.9.2, “Submit Network Job Command” on page 11-18. OfficeVision/400 uses SNADS processing. More information about SNADS and OfficeVision/400 is given in 13.5, “AS/400 Exchanging Distributions with Remote Systems” on page 13-34.

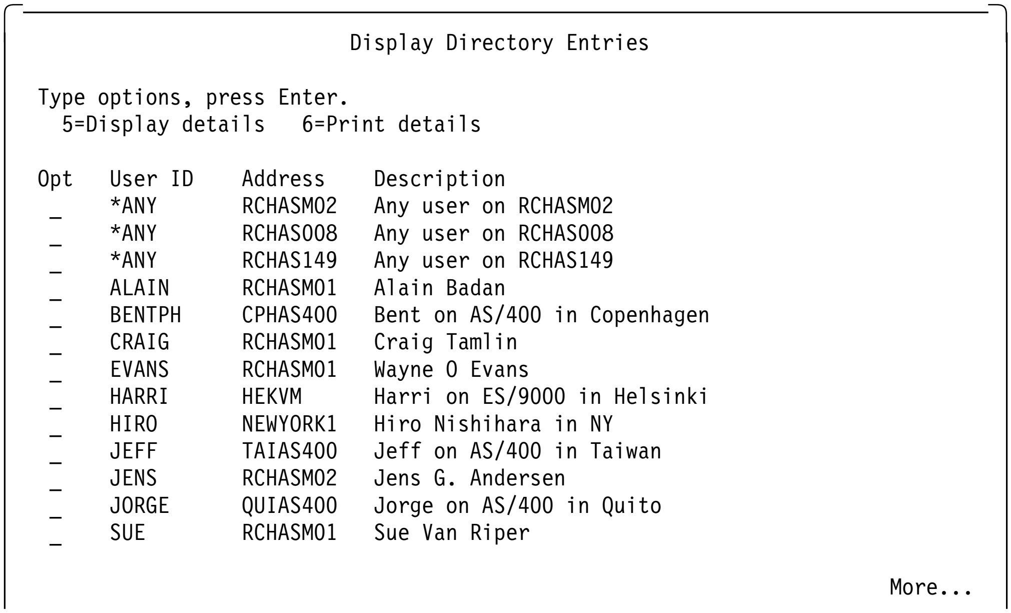
11.9.1 Configuring for SNADS

A SNADS configuration must exist on all systems that use the distribution services. Configuration involves line, controller, and device descriptions together with subsystem entries and establishing a *System Distribution Directory*.

SNADS is implemented as LU 6.2 applications and uses EVOKE security of *NONE. A default user profile must be defined on the target system in the SNADS subsystem (QSNADS). The supplied user profile is QSNADS and has a password of *NONE, preventing interactive signon.

System Distribution Directory

The system distribution directory must contain entries, either global or specific, for all the users who need to use SNADS applications and PC Support/400. It can be viewed using the DSPDIR (display directory) command. *SECOFR or *SECADM special authority is required to enroll users in the directory. Other users can only change their own entry, using the WRKDIR command. A user not enrolled in the directory is unable to perform any function with the directory. An example of the system distribution directory is shown in Figure 11-3.



```
                            Display Directory Entries

 Type options, press Enter.
   5=Display details   6=Print details

 Opt   User ID    Address    Description
  _    *ANY       RCHASM02   Any user on RCHASM02
  _    *ANY       RCHAS008   Any user on RCHAS008
  _    *ANY       RCHAS149   Any user on RCHAS149
  _    ALAIN      RCHASM01   Alain Badan
  _    BENTPH     CPHAS400   Bent on AS/400 in Copenhagen
  _    CRAIG      RCHASM01   Craig Tamlin
  _    EVANS      RCHASM01   Wayne O Evans
  _    HARRI      HEKVM      Harri on ES/9000 in Helsinki
  _    HIRO       NEWYORK1   Hiro Nishihara in NY
  _    JEFF       TAIAS400   Jeff on AS/400 in Taiwan
  _    JENS       RCHASM02   Jens G. Andersen
  _    JORGE      QUIAS400   Jorge on AS/400 in Quito
  _    SUE        RCHASM01   Sue Van Riper

                                                                   More...
```

*Figure 11-3. System Distribution Directory. Use the DSPDIR command to view the system distribution directory. The combination of user ID and address must be unique in the network. The user ID does not need to be the same name as the user profile. The address does not need to be the same as the system name (as defined in the network attributes), but could be a qualifier that might identify a department or user function. Both are defined using the WRKDIR command, where the actual system name and user profile are specified. There can only be one entry for each user profile. The entry *ANY RCHASM02 provides the capability to send and receive distributions for all users at the address RCHASM02.*

If there are many users in the SNADS network, this directory may have to contain entries to account for the users on each of their respective locations. Managing this directory is made simpler if global entries (*ANY RCHASM02) or (*ANY *ANY) are made. However, if the source system needs to be certain of the target (or destination) system of distributions, precise entries should be made. The distribution directory must be kept up to date when new user profiles are created by the system or network administrator, and also when users change locations or leave the business.

You may also use Directory Shadowing to automatically update the system directories in AS/400s in a network.

There is a full discussion of the System Distribution Directory in the *AS/400 Communications: Distribution Services Network Guide*.

11.9.2 Submit Network Job Command

The submit network job command (SBMNETJOB) provides the capability to send job streams to other systems. The target system controls how the job stream can be processed. This is determined by the network attribute *JOBACN*, whose values are shown in Table 11-3.

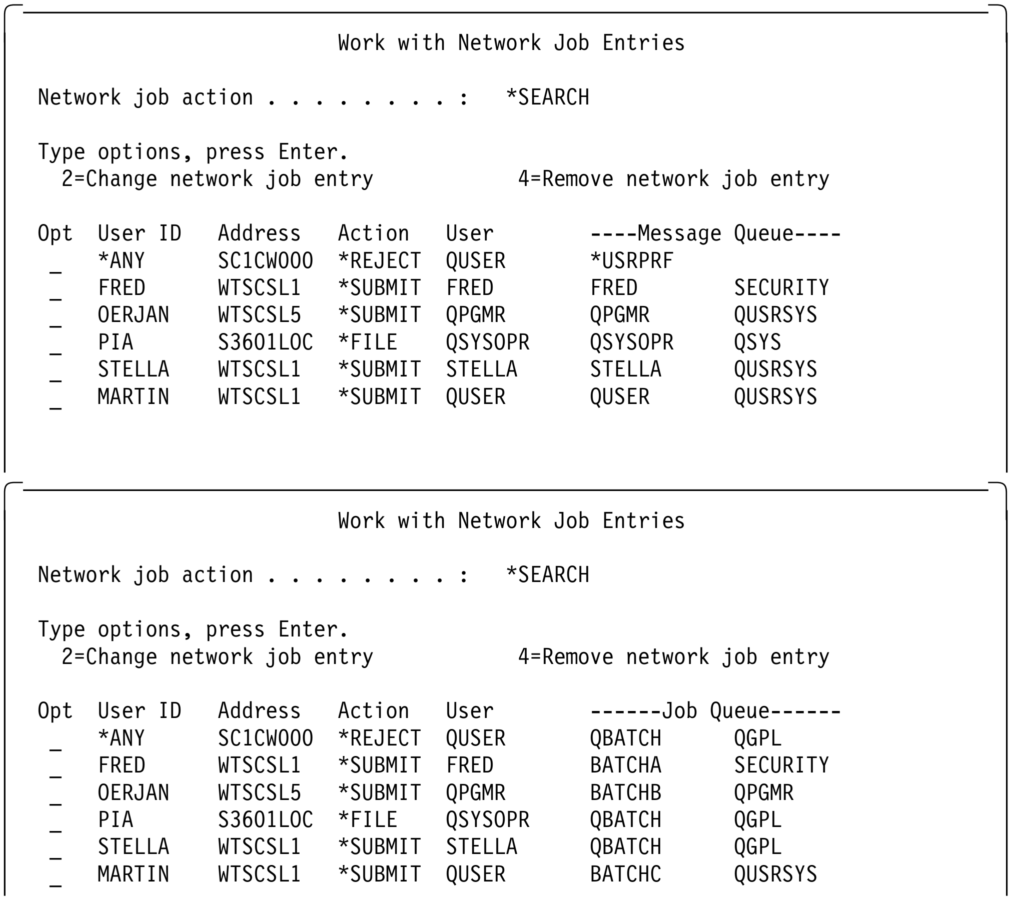
JOBACN value	***Meaning***
*REJECT	The input job stream is rejected by the system. This allows the target to secure itself from input streams received through the network.
*FILE	The input stream is filed in the queue of network files for the recipient. An authorized user may then view, delete, receive, or submit the job stream.
*SEARCH	The table of network job entries is searched to determine the action to take for the input job stream.

Table 11-3. Possible Values for JOBACN on the System Network Attributes

If *SEARCH is specified on the JOBACN parameter, the *Network Job Table* is searched for the action to take for the input job stream. It can be viewed using the WRKNETJOBE command. The ADDNETJOBE command is used to add entries into the network job table.

The target system network job entries will specify the user profile under which the job will run. For example, distribution directory update jobs could run under a user with administration rights. System operations, such as shutdown or network activation/deactivation, could run under the QSYSOPR profile.

An example of entries in a network job table is given in Figure 11-4.



```
                         Work with Network Job Entries

 Network job action . . . . . . . . :   *SEARCH

 Type options, press Enter.
   2=Change network job entry           4=Remove network job entry

 Opt  User ID   Address   Action   User        ----Message Queue----
  _   *ANY      SC1CWOOO  *REJECT  QUSER       *USRPRF
  _   FRED      WTSCSL1   *SUBMIT  FRED        FRED        SECURITY
  _   OERJAN    WTSCSL5   *SUBMIT  QPGMR       QPGMR       QUSRSYS
  _   PIA       S3601LOC  *FILE    QSYSOPR     QSYSOPR     QSYS
  _   STELLA    WTSCSL1   *SUBMIT  STELLA      STELLA      QUSRSYS
  _   MARTIN    WTSCSL1   *SUBMIT  QUSER       QUSER       QUSRSYS
```

```
                         Work with Network Job Entries

 Network job action . . . . . . . . :   *SEARCH

 Type options, press Enter.
   2=Change network job entry           4=Remove network job entry

 Opt  User ID   Address   Action   User        ------Job Queue------
  _   *ANY      SC1CWOOO  *REJECT  QUSER       QBATCH      QGPL
  _   FRED      WTSCSL1   *SUBMIT  FRED        BATCHA      SECURITY
  _   OERJAN    WTSCSL5   *SUBMIT  QPGMR       BATCHB      QPGMR
  _   PIA       S3601LOC  *FILE    QSYSOPR     QBATCH      QGPL
  _   STELLA    WTSCSL1   *SUBMIT  STELLA      QBATCH      QGPL
  _   MARTIN    WTSCSL1   *SUBMIT  QUSER       BATCHC      QUSRSYS
```

Figure 11-4. Example of Network Job Table Entries (WRKNETJOBE). The figure shows the two possible displays. Pressing F11 from the first display (showing the message queue used for the job) will show the Job Queue used for the job.

The figure shows that job streams, received from user FRED at location WTSCSL1, should be submitted using the user profile FRED, using the job queue BATCHA in library SECURITY.

This provides an excellent security measure for handling remotely submitted job streams. When a job stream arrives at the target system and the JOBACN parameter specifies *SEARCH, the job table is searched in the following order:

```
1. User profile    Address
2. *ANY            Address
3. *ANY            *ANY
```

This ensures that job streams from specific user profiles at a given location can be submitted using different rights than for other users at the same location.

Only entries from known users at specified locations should be allowed to be submitted automatically. Where the target system is less certain of the source location, the network job table allows the job stream to be received by a given user (*FILE), who may then examine the contents before deciding whether to submit the job. Where the target never wishes to receive job streams from a given source, then an entry such as the following could be used:

```
User ID   Address   Action
*ANY      SC1CW000  *REJECT
```

We do not recommend the use of an entry such as the following:

```
User ID   Address   Action   User
*ANY      *ANY      *SUBMIT  QUSER
```

When a job stream is received from a user at a location that does not have an entry in the network job table it will be rejected. The job stream may still be rejected if the job queue does not exist or if the user specified in the network job table is not authorized to the job queue.

11.9.3 Recommendations for SNADS

The following are recommendations for implementing good security for SNADS nodes:

1. Add communications entries to the QSNADS subsystem for the specific remote locations (or devices) from which SNADS distributions may be received.
2. Make specific (not generic or global) entries in the systems distribution directory for users who are to use the SNADS functions.
3. If job streams are not to be accepted from remote systems, change the JOBACN parameter on the network attributes to *REJECT.
4. If job streams can be accepted from remote systems, change the JOBACN parameter to *SEARCH. Make specific entries in the network job table for the acceptable users. If the content of the job stream and its source are in any way uncertain, specify *FILE. The job could still be submitted by user intervention once the job stream has been checked.
5. Maintain the system distribution directory to include only valid, current users. Update entries when users change departments or systems and remove entries for people no longer with the company.
6. See also 13.3.7, “Distribution Lists” on page 13-22 for more details.
7. QSNADS starts the APPC conversation with SECURITY(SAME) or SECURITY(NONE). Choose the one that is right for your network.

11.10 Transmission Control Protocol/Internet Protocol (TCP/IP)

There are many protocols that can be used to enable computers to share resources and transmit information across a network. *Transmission Control Protocol* (TCP) and *Internet Protocol* (IP) are two of the best known protocols. Since they are the most widely used, the term TCP/IP has become synonymous with a whole family of protocols.

The AS/400 implementation of TCP/IP includes:

- **File Transfer Protocol** (FTP) allows the user to log on to the remote system to *PUT* or *GET* files. Unlike the SBMNETJOB command (refer to 11.9.2, “Submit Network Job Command” on page 11-18), which allows a user to submit commands to be executed on the remote system, this function is only interactive. A user must have a valid user profile and password on both systems.

- **Simple Mail Transfer Protocol** (SMTP) is supported on the AS/400 using normal SNADS functions. It allows the sending and receiving of mail across a network. An AS/400 user can use OfficeVision/400 to handle mail. SMTP can also be accessed using the Send Distribution (SNDDST) and Receive Distribution (RCVDST) commands. Refer to Chapter 13, "OfficeVision/400" on page 13-1 for more details on OfficeVision/400 and 11.9, "SNA Distribution Services (SNADS)" on page 11-16 for SNADS.
- **TELNET Protocol** allows you to access and use the resources of a remote system as if your work station is locally connected to the remote system. The AS/400 TELNET support consists of two elements, ″TELNET client″ and ″TELNET server″, which means that users on the AS/400 can request access (client) to another system (server) and vice versa. The TELNET application forces a user to sign on as if at a local terminal.
- **Packet Internet Groper** (PING) used to verify a TCP connection to a remote system.
- **Application Program Interface** (API) are programmable modules that may be called from AS/400 Pascal. The APIs provide no security features; it is up to the applications using the APIs to provide security.

PING and the API are not discussed further in this section, but more information on these topics is given in the *AS/400 Transmission Control Protocol/Internet Protocol Guide*.

Note: There is also a redbook available, *AS/400 TCP/IP Configuration and Operation*, which contains additional information on TCP/IP.

11.10.1 Configuring for AS/400 TCP/IP

AS/400 TCP/IP is supported via Token-Ring, Ethernet, or X.25 line. You must create a line description depending on the type of network you are using. The controller and device descriptions can be automatically created for the TCP/IP jobs or configured manually. A subsystem, QTCP, is used for all the jobs associated with TCP/IP. A full description of setting up the environment for AS/400 TCP/IP is given in the *AS/400 Transmission Control Protocol/Internet Protocol Guide*.

11.10.2 TCP/IP Port Security

There are TCP/IP configuration commands (ADDTCPPORT and RMVTCPPORT) that restrict ports, so that only configured user profiles may use them. You can use this function when a server application has been developed that uses a specific port. The system administrator can prevent users from starting other applications that use the port.

11.10.3 TCP/IP File Transfer

The target AS/400 supporting TCP/IP must have a user profile created for each user who needs to send files from their system (AS/400 or non-AS/400) to the target system. When the file is sent, the sender specifies the name of the library that the file will be placed into. The library must already exist and the user must be authorized to it. The default library is the user′s current library. The user can issue the FTP subcommand ″CD″, (change directory), which changes the user′s current library.

Since the AS/400 supports files of fixed length and record size, the file being sent to the AS/400 must have a file length and record size no greater than the file on the AS/400. If the file does not already exist in the library, one will be created automatically. The file being sent will be created as a member of that file. Other files will be added as members of the file created. Table 11-4 on page 11-22 summarizes the results of attempts to send a file to an AS/400. Success is dependent upon whether the library, file, or members of the file exist and if they are compatibile.

Table 11-4. Consequences of Sending Files to the AS/400 Using FTP. AS/400 supports fixed length record files only. Files being sent must have file size and record length equal to or less than the existing file. The file being sent to the AS/400 will be received as a member of the AS/400 file.

Library Exists	File Exists	Member Exists	Replace Selected	Compatible Record Length(1)	Compatible File Size(1)	Result
Yes	Yes	Yes	Yes	Yes	Yes	Write data to member
Yes	Yes	Yes	No	N/A	N/A	Reject and send message
Yes	Yes	No	N/A	Yes	Yes	Create member, write data to member
Yes	Yes	No	No	No	No	Reject and send message
Yes	No	N/A	N/A	N/A	N/A	Create file; record length equals maximum record length of incoming file. Create member, write data to member
No	N/A	N/A	N/A	N/A	N/A	Reject and send message. Use the QUOTE CRTL subcommand to create a library on the remote AS/400

Note:

1. Applies when data is sent in stream transfer mode (as a stream of bytes). Does not apply when data is sent in image transfer mode (as a string of bits).

Although these are normal functions for FTP, they pose some security considerations for the AS/400.

Users already having a user profile on the AS/400 should be authorized only to the libraries and files they need to access. Allowing such users access via TCP/IP support should not present any additional security risk.

Users who need *only* to send files from other systems using TCP/IP but have no need for interactive sign on, still need to have a valid user profile and password on the AS/400. Such users should be LMTCPB(*YES). Menu security can be used to control the user should they attempt to sign on interactively, or use an initial program that signs them off.

Library and file authorizations are particularly important. By using FTP on other systems (AS/400 or non-AS/400) a user can retrieve a copy of a file member using the GET command. Read authority (*USE) is required for this function. TCP/IP users added to the AS/400 should have *EXCLUDE authority to all but the required libraries and files.

11.10.4 Simple Mail Transfer Protocol (SMTP)

Using the SMTP function of AS/400 TCP/IP, a user can send documents, notes or messages to another user. When using OfficeVision/400 to send mail to a host defined to SNADS as a TCP/IP host, the TCP/IP routines will automatically be used if the TCP/IP and SNADS subsystems are started.

Configuration of SMTP requires several steps, detailed in the *AS/400 Transmission Control Protocol/Internet Protocol Guide*. You can also use the AS/400 as an office gateway between SNADS and TCP/IP networks. In this situation you will have to configure your AS/400 for both environments.

Of particular importance is the updating of the system distribution directory and the SMTP host and alias tables. In simple terms, the host and alias tables contain the address and addressee information needed for sending and receiving distributions. The alias table is a nickname-type table, used to shorten lengthy addressee information or where special characters may cause a problem. Each SMTP user can have an alias table. Only the owner of the alias table and users with *SECADM authority are able to work with the alias table. Entries should only be made for known hosts and users.

11.10.5 TELNET Client

The AS/400 TELNET client allows an AS/400 TCP/IP user to sign on to and use the applications on a remote system that has the TELNET server application. It is up to the remote system (the server or target) to enforce security through normal user profile, password, and resource security measures.

11.10.6 TELNET Server

The AS/400 TELNET server allows a TCP/IP user on a remote TELNET client system to sign on to and run applications on the AS/400 system. With few changes to the system values, the TELNET server is automatically set up to support TELNET connections when TCP/IP is started. In 5250, 3270, or VT220** full-screen mode, the AS/400 automatically sends the AS/400 signon display when a TELNET connection is made.

Security Considerations: The security considerations are the same as for DSPT. Refer to 11.7.2, “DSPT Virtual Configuration Descriptions” on page 11-10 for this information.

Note: For a detailed description of how to configure, start, and end TELNET see the *AS/400 Transmission Control Protocol/Internet Protocol Guide*.

11.11 TCP/IP File Server Support/400

The TCP/IP File Server Support/400 product (5798-RYW) is an implementation of the Internet Protocol Standards (RFC-1094, RFC-1057, RFC1014) also know as the *Network File System* (NFS). NFS was originally developed by SUN** Microsystems and is based on a client/server model. The system that provides access to files is called a *server* (″target″ system) while the system that uses the service to access the files is called a *client* (″source″ system).

Many implementations of NSF allow one computer to act as a server in relation to a client while acting as a client in relation to another NFS server. The AS/400 TCP/IP FSS implementation is a server-only function, which means that the AS/400 system can act as an NFS server. FSS will allow access to both DLO and native database files on the AS/400 system. It will also allow users to store data on the AS/400 system that was created on other NFS client systems.

TCP/IP FSS will only run on top of the TCP/IP Connectivity Utilities/400 (5738-TC1) licensed program.

For more information on TCP/IP FSS/400 see the *AS/400 TCP/IP File Server Support/400 Installation and User′s Guide*.

11.11.1 File Server Support/400 Features

As mentioned earlier, the AS/400 can act as a NFS server in a network of UNIX** systems. On the AS/400 the files are contained in libraries and documents in folders while FSS works with a *directory* concept. FSS supports the exporting of the following AS/400 root (root means the highest level) directories:

- QDLS - AS/400 document library services hierarchical file system
- QSYS.LIB - AS/400 native database file system

The **QDLS File System** is a hierarchical tree structure, which is similar to the directory structure client users are used to work with. The QDLS file system supports *folders* and *documents* on the AS/400.

Note: A folder is similar to client directories and documents are similar to files.

The *QSYS.LIB Directory* contains native AS/400 database files, which have a different organization than client users file systems. The following extensions are used in the QSYS.LIB file system:

- LIB defines a library subdirectory
- LF identifies a logical file
- PF identifies a physical file
- FILE identifies a source file subdirectory
- MBR identifies a file within a source file subdirectory (that is, FILE)

Note: You can view QSYS.LIB as a subset of QSYS as FSS only can access *files* (physical, logical, and source) on the AS/400.

The AS/400 system has no ″universal″ root directory that covers both (DB files and documents) file systems.

Mounting the Directory. Client systems access a directory that is physically located on a remote server (the AS/400, for example) by mounting the directory. Once mounted, the users in the client system can access this directory in the same way as if it were residing on their local system.

Note: There is no similarity in the ″native″ AS/400 world to *mounting* a directory, but using AS/400 terminology it will be a function that permits users on a source system to access a selection of files in a library or documents in a folder on a target system, just like their own local files and documents.

11.11.2 FSS/400 User Profiles

To use FSS/400, you must create two special user profiles and also add them in the system directory to access the document library services (QDLS):

- *Q7FSOWN*. This profile must have *SECOFR authority. The file server is using this profile. Q7FSOWN is the user profile associated with *root* users. The root user is a UNIX user with authorization similar to QSECOFR.

- *Q7FSUSER*. This profile is used when the AS/400 gets a request for processing files from a user who has not been defined (that is, has no user profile on the AS/400). Q7FSUSER will have authorization similar to the QUSER user profile.

When FSS/400 receives a request from a NFS client, only the UNIX User ID (UID) is known. These UIDs must be mapped to AS/400 user profile names. You can use the command WRKFSSUSR and use the display presented to map (add, change, remove) AS/400 user profiles and UIDs.

11.11.3 Recommendations for FSS/400

The FSS user profiles on AS/400 should have password *NONE. This will prevent the use of the FSS user profiles for interactive work. Since Q7FSOWN must have *SECOFR authorities you cannot restrict its access to objects.

The NFS client system often use UNIX permissions or authorizations, which are different from the AS/400 authorizations. In a UNIX system, a user can access a file with the following permissions:

- read (r) - the user can read the contents of a file
- write (w) - the user can write new data to the file and the user can delete the file
- execute (x) - the user can execute a program if the file contains executable code and the user can search if it is a directory
- none (-) - the user is not authorized to the file

UNIX has nothing similar to AS/400 operational rights and the UNIX permission scheme does not have the same granularity as the AS/400 object authorization. You should be aware of the differences and the risk of giving to much authority to NFS client users because of the limited options in UNIX authorization.

For more information on UNIX permissions and the mapping of AS/400 object authorities to UNIX see the *AS/400 TCP/IP File Server Support/400 Installation and User's Guide*.

The redbook *IBM AS/400 as a Network File Server* GG24-4092, contains additional information about AS/400 security in an NFS environment.

11.12 AS/400 3270 Device Emulation

The AS/400 3270 Device Emulation is provided as part of OS/400. To the host, the AS/400 system appears to be a 3274 control unit with attached display stations (3270-type) and printers (3287/3289). This enables the user to access applications on ES/9000 or other systems that support the 3270 data stream.

The users are treated as ordinary work station users at the target system. The security is the responsibility of the target system. However, a possible exposure arises whereby the AS/400 3270DE user could sign on to the ES/9000 system and send job streams back to the AS/400. You may want to limit the authority to the STR3270EML command.

Note: More information on 3270DE, configuration, and usage is available in the *AS/400 Communication: 3270 Device Emulation Guide*.

11.12.1 Job Streams

Communication could be set up in a way that, for example, allows a VM user to transfer files to the AS/400. The file content can be a job stream to be executed on the AS/400.

The AS/400 can only be configured in a way that allows the job, arriving as a VM file, to be started with some intervention on the AS/400. When a file is sent from a VM system, it is received as a *Network File* and is placed on the *Network File Queue*. An authorized user must receive the file or submit the job stream to a job queue.

The user in the job description used determines the authorities of the job. If the job stream does not specify an AS/400 job description, the system will default to the QBATCH job description, which specifies a default user QPGMR, under which the job will be executed. It is not possible to specify a job description in the submit job command. It must be specified in the *//BATCHJOB* statement of the file that is to be executed. It is not possible to specify a job description that specifies USER(*RQD).

It is possible to create a CL program that executes on the AS/400 as a ″never ending program″, to automate the receiving of the file and submitting the job. In this case, batch jobs might be started from other systems, by users that not are supposed to use this function. Refer to the redbook *VM-AS/400 Connectivity and Functional Use* (GG24-3430), for a description of such a program. The use of such programs is not recommended, unless the target AS/400 is always certain of the source and nature of the batch job stream.

This is in contrast to sending job streams in files between AS/400s, which is discussed in 11.9.2, “Submit Network Job Command” on page 11-18.

11.13 Remote Job Entry (RJE)

AS/400 RJE is part of the *Communications Utilities Licensed Program*. The primary purpose of RJE is to submit jobs to, and receive output from, a host (ES/9000) system. You can use the processing power of the host system, while you still use your own local data and applications. RJE is the preferred method for doing this, as it provides file input and printer control. Your AS/400 must be connected to a host system either through a communications line (SNA or BSC) or a local area network.

RJE is for *batch only* functions and all the processing of the submitted data takes place on the host system.

11.13.1 Recommendations for RJE

There is no interactive signon to your system, and the applications you are able to submit for processing on the host, is controlled by the host.

To start an RJE session, you must have the required authorities to the objects referenced by the Start RJE Session (STRRJESSN) command.

RJE has no special security features. When the output data from the host is received on the AS/400 it has a public authority just like any other object. If you have special security requirements, such as confidential print log data sent from the host, there is an IBM-supplied program QMRSWTR that can be used.

You can call QMRSWTR from a user program. By specifying the correct parameters and overriding the output file (printer or DB) you can direct the data to a secured file in a secured library. You also have the possibility to pass RJE data from the host directly to an application, using QMRSWTR.

For more information on RJE and QMRSWTR see the *AS/400 Communications: Remote Job Entry (RJE) Guide*. It contains detailed information on installing, configuring, and using RJE.

11.14 Electronic Customer Support (ECS)

Your AS/400 has an integrated set of functions to help service and support your system:

- HW and SW problem analysis, reporting, and management
- Copy screen image
- Question and answer support
- IBM technical and product information access

We call these functions *ECS Functions* and they are provided as part of the OS/400 product.

The necessary ECS definitions is normally part of the HW installation process. The AS/400 communications attachment and modem used for ECS is also included for the major part of the AS/400 models. For more information on ECS, see the *AS/400 System Operator′s Guide*.

11.14.1 Recommendations for ECS

ECS normally uses a switched V.24 connection with automatic call when you connect your system to IBM to request PTFs or report problems. There is no security exposure as you are initiating the call.

ECS also has a function called *Start Remote Support* (STRRMTSPT). It creates and varies on all the configuration objects needed for remote support (that is, a support organization can get access to your system from a remote work station). You must provide a telephone number, user profile, and password before the support person can sign on to your system. Once the support person is signed on he or she has all the rights that the user profile grants.

You should verify the identity of the support personnel (you could make a ″call back″) before you supply a user profile and password. The default user profile of STRRMTSPT is QPGMR.

You can create a user profile (for example, RMTSUP) with only the authority needed to access what is necessary, and change the password on the profile to *NONE, or disable it, after use. You must supply a new password, or enable the profile, before the next signon request.

You can also use the audit journal (see Chapter 6, “Audit Journal” on page 6-1) to monitor the work done by the remote support user profile.

Also remember to disconnect from technical support (ENDRMTSPT) after the work is done. ENDRMTSPT will ″vary off″ and delete all the configurations objects that were created by STRRMTSPT.

11.15 User-Written Applications and File Transfer Support

This section covers considerations for implementing security in user-written applications.

User-written communications applications can be written using any of the following facilities:

- Intersystem Communication Function (ICF)
- Common Programming Interface for Communications (CPI-C)
- CICS/400*
- OSI
- TCP

11.15.1 Intersystem Communications Function

An *Intersystem Communication Function File* is a device file for communications. It is created from Data Description Specifications (DDS) source, and contains the formats for the data to be sent or received by applications over a communications link. An application will reference an ICF file (for instance by the INPUT statement for RPG).

ICF presents a common application interface for the communications facilities available on the AS/400 (APPC, SNUF, asynchronous, bisynchronous, Retail, Finance, and Intra). ICF allows the application programmer to define the application data externally to the program and independently of the protocol type. An overview of the process involved in user-written applications is included in Appendix F, "User Communications Application Programming Steps" on page F-1 and can be used in conjunction with the following discussions.

The link between the application program and the physical communications is made, by a *Device Entry* which must be made in the ICF file, using the ADDICFDEVE command. This entry is a program device entry containing the remote location name for the target system. A program will *ACQUIRE* a particular program device for the type of communications that the application is to use.

If the communications type for the application is changed, the application does not need to be changed; rather the program device entry can be changed (and possibly the LIND, CTLD, and DEVD dependent on the nature of the new communications type). The DEVD for the communications configuration with the same remote location name provides the link to the physical communications. The ICF file allows the definition of program devices for different communications types.

Note: The types of communications supported are APPC, SNUF, BSCEL, asynchronous, intrasystem, Finance, and Retail Communications. See the appropriate communications manuals of these products for full details.

The program devices specified in one file can be for different communications types.

ICF File Security

The ICF file is subject to normal AS/400 object authorization. Since different communications types may be used for different applications, the same ICF file should not be used for applications that may have different security requirements. Similarly, the same application may need to use different types of communication links. Authorizations to the ICF file should be granted on the basis of the application sensitivity.

Security is specified in the DDS for the ICF file and can have one of the following values:

- SECURITY(NONE) - the default, or specify SECURITY(3 *NONE)
- SECURITY(SAME) - specify SECURITY(3 *USER)
- SECURITY(PGM) - specify SECURITY(1 profile name, 2 password, 3 user ID)

Security Information in User Applications

A user application can send one of the following with the EVOKE command:

- Only a user profile and Already Verified Indicator if the bind is secure (only valid for APPC)

or

- Both a user profile and a password

or

- No security information

non-secure location the default target user profile will be used, unless both a user ID and password are sent. If the source system is defined as a secure location the default user profile will be used, and neither a password nor an AVI indicator is sent with a user ID.

If it were satisfactory for every user at a particular site to appear to the network as the same user, it would be possible to code applications with a literal in the user profile field and then define one user profile on each system to represent each possible communications partner system. Since it is not possible to access a user′s password from the security file, it must be either hard-coded or prompted for in the application. *We do not recommend the use of hard-coding the password for AS/400 applications, since this presents a severe security exposure for the target AS/400*.

In a large network where program-to-program communications is important, locations should be defined as secure and programs coded so that only the user profile and AVI are sent over the network when using APPC. This will at least reduce the maintenance problem of keeping passwords in synchronization across many systems, and it will allow logging of user actions to provide accurate information as to who is the real user. Refer to the *AS/400 Communications: Intersystem Communications Functions Programmer′s Guide*.

File Transfer Support (FTS)

AS/400 File Transfer Support is a function of OS/400 that can be called by a user-written application. It allows the application program to send database file members to another AS/400 (or S/36) and retrieve database file members from an AS/400 (or files and library members from a S/36).

FTS applications always send the requesting user profile and a password for the remote system, regardless of whether a location is secure or not. The password is often obtained from a prompting screen format when run interactively, or coded in the program if FTS is to run in batch. If a file is created on the remote system the requesting user profile becomes the owner of the file on the target.

11.15.2 CPI Communications (CPI-C)

The CPI Communications (CPI-C) provides an interface to APPC communication. Though LU6.2 defines three kinds of security for the ALLOCATE and the CM_ALLOCATE verbs, the CPI-C architecture defines four of them:

- SECURITY(NONE): Security information is not sent when a program start request is sent to a target system
- SECURITY(SAME): The calling user ID and AVI is sent to a target system along with the program start request
- SECURITY(PGM)
- SECURITY(PROGRAM STRONG)

CPI-C Level 2 has four new security calls. They are not *fully* supported by AS/400:

- **CM_SET_CONVERSATION_SECURITY_TYPE** allows the user to choose the security level necessary for the application. The security types supported include:
  - CM_SECURITY_NONE:

    No access security information is included in the conversation startup request.
  - CM_SECURITY_SAME:

    If the remote system allows local verification of the password (SECURELOC(*YES)), APPC sends the user ID and AVI.

    If the remote system does not allow local verification of the password (SECURELOC(*NO)) and password protection is active, Enhanced SECURITY(SAME) is used. In other words, the password for the user is retrieved and a protected password is sent.

    If the remote system does not allow local verification of the user SECURELOC(*NO)), and the password protection is not active, APPC does not send any security information. Therefore, the security level is SECURITY(NONE).
  - CM_SECURITY_PROGRAM:

    The user ID and password specified by the program are sent with the conversation startup request. The program specifies the user ID by using the Set_Conversation_Security_User_ID (CMSCSU) call. The program specifies the password by using the Set_Conversation_Security_Password (CMSCSP) call.
  - CM_SECURITY_PROGRAM_STRONG:

    The user ID and password specified by the program are sent with the conversation startup request. The program specifies the user ID and password using the CMSCSU and CMSCSP calls, as for CM_SECURITY_PROGRAM. The conversation security type ensures that a protected password is sent. If a protected password cannot be sent, the Allocate request fails.

- **CM_SET_CONVERSATION_SECURITY_USER_ID** to be used together with CM_SECURITY_PROGRAM or CM_SECURITY_PROGRAM_STRONG
- **CM_SET_CONVERSATION_SECURITY_PASSWORD** to be used together with CM_SECURITY_PROGRAM or CM_SECURITY_PROGRAM_STRONG
- **CM_EXTRACT_SECURITY_USER_ID** to extract the security user ID

All security specifications must satisfy the requirements of the target system.

11.15.3 CICS/400

There is no special security facilities for CICS/400 on AS/400. Standard security measures apply.

CICS/400 always uses conversation level SECURITY(SAME).

Command-level security, which restricts the use of system programming commands such as EXEC CICS INQUIRE, EXEC CICS SET, EXEC CICS PERFORM, and EXEC CICS DISCARD are not provided by CICS/400. The use of these commands may be controlled by resource-level security on the programs that use the commands.

11.15.4 Open Systems Interconnection (OSI)

There are security services defined in the OSI basic reference model. They are designed to provide security functions by defining security services and mechanisms in each protocol layer, and described in the standard, *Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture, October 1988*. Since the most of the security services in the reference model are still under development in ISO (International Organization for Standardization) and not yet standardized, the OSI Communication Subsystem/400 and the OSI File Services/400 have only a partial implementation of Access Control. Therefore, security of AS/400 OSI support must be considered mainly in the application layer, except for configuration security.

You must have an authority higher than QPGMR to add, change, and remove objects required for the OSI support. If you let the system automatically configure a controller and device description for OSI, you should ensure that the user profile QOIUSER has an authority for these descriptions.

OSI Communications Subsystem/400

The OSI Communications Subsystem/400 provides communication interface for the OSI network. Presently, there are three kind of interfaces:

- User application programs
- OSI Message Services/400
- OSI File Services/400

The OSI/CS configuration commands (ADD, CHG, RMV) requires QPGMR authority or higher.

User Application programs

User application programs are allowed to have access to ACSE (Association Control Service Element) presentation and session layers through a programming interface provided by the OSI Communications Subsystem/400. There are quite a few callable services and they have an *USE authority for *PUBLIC users. Any user can use application programs with those callable services. When you want to limit the use of this interface to some users, the only thing you have to do is limit the *USE authority for OSBAEE. The interface program OSBAEE must be called in all user application programs to build an application entity environment. Without this process the user programs cannot run.

OSI Message Services/400

The OSI Message Services/400 can send notes, documents, messages, save files, spool files, database files, and job streams to other systems through the OSI network. No user can send or receive these objects unless they are registered by the administrator in the directory with their originator/recipient (O/R) name. This does not mean that the OSI Message Services/400 provides any protection on those objects by itself. All objects should be secured using the facilities in OS/400. For example, you can prohibit some users from sending data files to other system by revoking their authority for the files, even if they are authorized to use the SNDDST or the SNDNETF command. Therefore, you must follow the security considerations described in Chapter 13, “OfficeVision/400” on page 13-1 and 11.9, “SNA Distribution Services (SNADS)” on page 11-16 to properly secure those objects.

From a configuration point of view, the administrator needs to know the password for each remote Message Transfer Agent (MTA), which however has nothing to do with end users.

The OSI/MS configuration commands (ADD, CHG, RMV) requires QPGMR authority or higher.

OSI File Services/400

The OSI File Services/400 provides its own security mechanisms to protect some resources:

- User password
- Filestore password
- Filestore create password
- File access permission and password

User Password: No user can have access to the OSI File Services/400 unless they are registered as a user by the OSI File Service/400 administrator. A user password can be defined when a user is registered. Though the registration of user is mandatory for every user to start the file services (STROFS), a user password is required only for using the application programming interface (API) to have access to OSI File Services/400. A password is not required for users to use its interactive interface. In other words, you have to provide your own interface program using APIs, and restrict the use of the program. In this way you can limit the access to the OSI File Services/400 in addition to using a password. When you register a user for the OSI File Services/400, you can define whether or not the user can have access to his own and other users′ filestores. Though a user password can be changed only by the user through an interactive interface, only the OSI File Services/400 administrator or a privileged user can retrieve and display the password.

The password security for a filestores is optional. There are two kinds of passwords that can be defined for a filestore:

- A filestore password
- A filestore create password

Each user can specify these passwords for his filestore when he adds it in the local resource directory (LRD).

Filestore Password and Filestore Create Password: The filestore password is used to limit access to the filestore. Unless you specify a filestore password, the filestore can be accessible by any local and remote users who are given a right to have access to another user′s filestore. The filestore create password is used to restrict creation of new files. If you do not specify the filestore create password any user with access to the filestore can create a new file.

Every user can also define who is allowed to have access to his filestore when the filestore is added to the LRD. He can grant authority to the filestore to three types of users: the owner, other local users, and remote users.

Action Permission: When you create a new file in your own or another user′s filestore, you can limit the actions on the file such as read, read attributes, extend, change attributes, replace, and delete by negating each action. When you permit an action on a file, you can optionally specify an access password so that only those users who know the password for each action can process the file. If you permit an action and specify no access password any other user can process the file with the action permitted. When a file is sent to a remote system as a new file, these attributes of action permission and associated passwords are inherited by the remote file. Also, if the remote system is an AS/400, the remote file is given *ALL authority to the addressee user and *EXCLUDE authority to *PUBLIC users. Since the access password can be displayed only by the owner of the filestore, you must remember the access password for your file created in other users′ filestores.

Note that all of the password security functions stated above are local functions within the OSI File Services/400. They do not substitute or override any level of security provided by OS/400. Thus, you have to have proper authority to have access to a file even if you are permitted to perform some operations on the file by the OSI File Services/400.

To summarize:

- APIs and Interactive Interface exist
- One user profile is designated as the *Privileged User*. Each user must be authorized by the Privileged User.
- Each user has a password for accessing OSI/FS via the API
- Each user has *one* filestore
- Each user must be authorized to use:
  - His/her own filestore
  - Other local filestores
- Each filestore has two passwords:

  - One to access the filestore
  - One in order to create files in the filestore

- Each filestore has access rights for:

  - The owner
  - Other local users
  - Remote users

- Each filestore may have multiple files

- Each file has a list of permitted actions:

  - read, extend, replace, delete
  - read-attributes
  - change-attributes

  Each action also may have a password

11.16 Pre-start Jobs on the Target System

Once the physical communications link has been established between source and target systems, the user application program must issue an *ACQUIRE* with the program device to start a communications session. The program on the remote system is started by the *EVOKE* for ICF and *ALLOCATE* for CPI-C applications. The EVOKE or ALLOCATE commands may also send security information with the program start request.

Note: The program itself can vary on the source configuration descriptions; the target system configuration descriptions must be started from the target system.

In order to minimize the time to start the program, it is possible to have the program start automatically when the subsystem used for the communications is started. This is achieved by having a *Pre-start Job Entry* in the subsystem.

The pre-start job entry contains the name of the program to be started and the name of the user profile used for the job. When the pre-start job is started, authority checking for all the objects needed by the job is done against this user profile. When the EVOKE or ALLOCATE is issued by the program on the source system, the program start request attaches to the pre-started job. The user profile (sent from the source or the default user profile on the target) is only checked against the authorities to the target device, the program, and its library. If this user profile does not match the authorities to the objects to which the pre-start job user profile is authorized the target program cannot be started.

The user profile for the pre-start job should have authority only to those objects needed for the program. It should be a user profile that is used solely for this purpose. In this way the user profile matching for the pre-start job and the program start request does not become an exposure.

11.17 Summary for AS/400 Communications Security

This chapter has covered the key elements that should be considered when securing an AS/400, as both the target and source when communicating with other systems and users. In addition to implementing the normal AS/400 resource security, good security for communicating AS/400s can be achieved by use of AS/400 Work Management features (subsystems, communications entries, jobs and job descriptions and so on). Of particular importance is the user profile that is used on the target system.

Chapter 12. PC Support

This chapter discusses the IBM PC Support/400 licensed program product (5738-PC1), from a security, and auditing perspective. It refers to PC Support running in the IBM PC/DOS environment, and also OS/2.

In V3R0M5 and V3R1, the PC Support product is replaced with a product family called Client Access/400. The product family now contains separate products for each of the client environments and the server code is now a part of OS/400. The following is the mapping of the PC Support product options to the new product family as seen from the Display Software Resource (DSPSFWRSC) command:

V2R3M0 Product (option)	V3R1M0 Product (option)
5738-PC1 (BASE) Host Server	5763-SS1 (12)
	5763-XL1 Client Access/400 for DOS
(1) DOS SBCS	(1)
(2) DOS DBCS	(2)
	5763-XF1 Client Access/400 for OS/2
(3) OS/2 SBCS	(1)
(4) OS/2 DBCS	(2)
(10) RUMBA OS/2 SBCS	(3)
(11) RUMBA OS/2 DBCS	(4)
(12) Comm Mgr/400	(5)
	5763-XB1 Client Access/400 for DOS with Extended Memory
(5) Ext DOS SBCS	(1)
(6) Ext DOS DBCS	(2)
(8) RUMBA SBCS	(3)
(9) RUMBA DBCS	(4)
	5763-XA1 Client Access/400 Family - Base
(7) PC Tools Folder	(1)

All references in this chapter to PC Support are also valid for the Client Access products listed above (the original clients). The information may not be correct for the new clients and servers that were announced for V3R1 (5763-XC1 Client Access for Windows 3.1 and 5763-XG1 Client Access for OS/2 2.1).

12.1.1 Introduction

PC Support/400 is a co-operative processing application, and as such has components in both the AS/400 and in the personal computer (PC). The PC Support product uses the AS/400 security functions to determine access to the AS/400′s objects. You cannot override the security of the AS/400 by using PC Support, but you have to be aware of how the IBM PC Support/400 program utilizes the AS/400′s resources.

From a security point of view, there are three primary areas of concern:

- How to prevent unknown PCs from connecting to an AS/400.
- How to protect the data in an AS/400 from unauthorized access or manipulation. PC Support users may be able to reach objects in the AS/400 system that are unavailable to an ordinary user. This can be an exposure that has to be addressed.
- How to prevent the spread of PC viruses via an AS/400 to other PCs. Viruses represent a high exposure to PC data integrity, and PC Support/400 could propagate these viruses over a network.

12.1.2 IBM PC Support/400 Functions

PC Support provides the means to access and use AS/400 resources with the flexibility and ease of use of the personal computer. Functions include:

PC Support Function **Description**

Work Station Function Allows a PC DOS user to work on an AS/400 as if using an ordinary work station. OS/2 users access this function from the Communications Manager 5250 Workstation Emulation Feature Programs. (Refer to "OS/2 Special Security Considerations" on page 12-3). Windows users have access to this via the Rumba option.

Printer Emulation for Work Station Function
Allows a PC user to use a PC printer as an AS/400 printer.

Virtual Printer Function Allows a PC user to use AS/400 printers as PC printers.

5250 Session Manager Allows a PC user to view up to five display or printer sessions at the screen simultaneously, through windowing.

PC Organizer (PCO) Provides a single menu for both PC and AS/400 options. PCO is basically a function that enables PC commands to be issued from an AS/400 menu.

Shared Folders Function Allows a PC user to:

- Store PC information, such as programs and files, in shared folders in AS/400.
- Access PC files and programs that are stored in shared folders in AS/400.

Message Function Allows a PC user to send messages to and receive messages from other PCs or workstations.

Transfer Functions Allows a PC user to transfer data from and to an AS/400.

Submit Remote Command Function
Allows a PC user to send commands from the personal computer to an AS/400 system to be executed.

Data Queue Support Allows PC programs and OS/400 programs to communicate using data queues.

Remote SQL Allows PC applications to run SQL statements in an AS/400 system.

An extremely important feature are the *User Exit Programs*. These programs may be used to ensure that your data is both secure and correct. An exit program may be used to determine if users have authority to perform a particular operation.

12.1.3 PC Support Connection

To run IBM PC Support/400, a special PC Support program called the *Router* must be installed on the PC. The interface on the AS/400 side is a communications program. Communications from a PC to the AS/400 system uses Advanced Program to Program Communications / Advanced Peer to Peer Networking (APPC/APPN) support. You can find more information about APPC/APPN in Chapter 10, “Communications Security in SNA” on page 10-1.

The PC Support router provides support for several different types of communication attachment interfaces between your personal computer and the AS/400. The different PC Support router connectivities are:

- Twinaxial, either local or through cluster controllers:
  - 5394 Remote Workstation Controller
  - 5494 Remote Workstation Controller
- Local Area Network connections:
  - Token Ring
  - Ethernet
- Synchronous Data Link Control (SDLC)
- Asynchronous connection (only under DOS, not under OS/2)
- X.25 Connection (only under OS/2, not under DOS)

The PC Support router program controls the communications between one or more IBM PC Support/400 functions in the personal computer and their counterparts in AS/400 systems.

The main advantages of this router design are:

- Single router for all available connections
- Single emulator (work station function or OS/2 5250 workstation emulation feature)
- All functions supported in all environments
- Multiple connections

OS/2 Special Security Considerations

When using OS/2, you have different products which enable you to connect to the AS/400, such as:

- OS/2 Extended Services Communications Manager
- Communications Manager/2
- Communications Manager/400

These products provide Advanced Program to Program Communications (APPC) services to OS/2 PC Support when communicating with the AS/400. Extended Services Communications Manager and Communications Manager/2 also provide 5250 emulators. Note that Communications Manager/400 does not provide a 5250 emulator, but you can use any emulator running in a Virtual DOS Machine (VDM), for example Rumba/400.

All the security considerations discussed in this chapter apply to both DOS and to any of the three Communications Manager products, as all of them use APPC to communicate to the AS/400.

Unless there are special considerations, we will refer to all of the OS/2 Communications Manager Products collectively as ″Communications Manager″.

12.2 Using Security System Values with PC Support

The AS/400 security system values have the overall control of the security in your system. If you want to know more about security system values, see Chapter 2, “System Values and Network Attributes” on page 2-1. You will also find more information in the *AS/400 Security Reference*.

The following system values need to be considered when installing and using PC Support:

QDSPSGNINF Determines whether the signon information display is shown after signon using the PC Support Work Station function or the OS/2 5250 Workstation Emulation Program. We recommend value 1, so users can monitor invalid attempts of accessing their profiles, and be informed when a new password is required.

Note: In V3R1 QDSPSGNINF is valid for the STARTRTR command as well.

QINACTITV Specifies, in minutes, how long a job (in this case a PC Support job) is allowed to be inactive. A job is considered to be inactive when it doesn′t respond to a message (its status is DSPW or MSGW). When the value specified in QINACTITV has been reached, the system takes the action specified in the QINACTMSGQ system value. QINACTITV only has effect over the PC organizer function and the Work Station function. All the other PC Support functions are not affected by this system value. For example, if you set QINACTITV to 20 and start the file transfer function, this could be inactive (no interactions with the AS/400) for more than 20 minutes, however, the system will not take any action. You should be cautious of this, because it could represent a security exposure. Refer to 12.9.1, “Keyboard Lock” on page 12-26.

QINACTMSGQ Specifies what action the system takes when the inactive job time out interval (specified in QINACTITV) has been reached. You can specify *ENDJOB (which is the default value), *DSCJOB or specify a message queue name. A user program must monitor this message queue and take action as needed. In this way you can make decisions about particular devices or user profiles. This is the method we recommend when you use PC Support. You can find an example of a program that monitors a

QINACTMSGQ message queue in figure Figure 12-1 on page 12-6.

QLMTSECOFR Controls whether a user with all-object (*ALLOBJ) or service (*SERVICE) special authority can use any device. If this system value is set to 1 all users with *ALLOBJ or *SERVICE special authorities must have *CHANGE authority to use the devices. We recommend you to set this system value to 1.

QMAXSIGN Controls the number of consecutive invalid signon attempts by local and remote users. Once the QMAXSIGN value is reached the system determines the action to be taken using the QMAXSGNACN system value.

There is a special consideration you must be aware of. If the QMAXSGNACN is set to ′1′ (which is ″vary off device″) the QMAXSIGN value virtually will have no effect on the number of consecutive times a user can enter a invalid password when starting the router (STARTRTR) function. This is because the user is actually starting an APPC communications request, and the system will not disable the APPC device.

It is a security exposure. If you cannot set QMAXSGNACN to 2 or 3 you may choose to prevent further use of that APPC device description doing the following:

- Use the QSYSMSG message queue to monitor special system messages, such as the message that is generated when this condition occurs (CPF1269). If you want more information about QSYSMSG message queue, refer to Appendix A, “QSYSMSG Message Queue” on page A-1.
- Record the attempts. It should be reviewed by the person responsible for security in AS/400.
- Issuing the End Mode (ENDMOD) command to set the allowed jobs to zero. This allows jobs that are currently using the APPC device description to remain active, but prevents other jobs from starting until the condition is understood.
- Counting the number of attempts in a given period. You could establish a threshold and check this value in a program that monitors the QSYSMSG message queue. When this value is reached, take any serious action (such as changing the maximum number of sessions to zero).

  This function can also be provided with IBM Security/400, using the audit journal. Refer to Chapter 15, “IBM Security/400” on page 15-1 for more information.

QMAXSGNACN Determines what the system does when the maximum number of signon attempts is reached at any device. You can specify to vary off the device, disable the user profile or both. We recommend that you disable the user profile and vary off the device.

QRMTSIGN Specifies how the system handles remote signon requests. PC Support Workstation Function is actually a remote signon request. Refer to 12.4, “PC Support Functions from a Security Point of View” on page 12-11 for further explanation.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...
  100 PGM
  200 /*************************************************************/
  300 /*PROGRAM: INCTPGM                                            */
  400 /*PURPOSE: THIS PROGRAM MAY BE USEFUL TO MONITOR INACTIVE JOBS */
  500 /*         THE PROGRAM HAS TO BE SUBMITTED AS A BATCH JOB     */
  600 /*         AND IT WILL LOOP UNTIL SOMEONE STOPS IT (ENDJOB)   */
  700 /*         BEFORE YOU RUN THE PROGRAM, THE MSGQ               */
  800 /*         QGPL/INACTMSGQ MUST BE CREATED.  WHEN THE MESSAGE  */
  900 /*         CPI1126 ARRIVES ON THE MSGQ, THE PROGRAM WILL SEND */
 1000 /*         A MESSAGE TO THE JOB USER IN THE MESSAGE DATA.     */
 1100 /*         THIS PROGRAM ASSUMES QINACTITV IS SET TO 20.       */
 1200 /*************************************************************/
 1300            DCL       VAR(&MSGQ) TYPE(*CHAR) LEN(10)
 1400            DCL       VAR(&LIB) TYPE(*CHAR) LEN(10)
 1500            DCL       VAR(&MSGID) TYPE(*CHAR) LEN(7)
 1600            DCL       VAR(&MSGDTA) TYPE(*CHAR) LEN(100)
 1700            DCL       VAR(&JOBNAME) TYPE(*CHAR) LEN(10)
 1800            DCL       VAR(&USERNAME) TYPE(*CHAR) LEN(10)
 1900            DCL       VAR(&JOBNBR) TYPE(*CHAR) LEN(6)
 2000
 2100 /*  RECEIVE THE MESSAGE AND ITS DATA                          */
 2200
 2300  START:     RCVMSG    MSGQ(QGPL/INACTMSGQ) WAIT(*MAX) RMV(*NO) +
 2400                         MSGDTA(&MSGDTA) MSGID(&MSGID)
 2500
 2600            IF        COND(&MSGID *NE 'CPI1126') THEN(GOTO +
 2700                         CMDLBL(START))
 2800
 2900 /*  IF THE MESSAGE CPI1126 HAS ARRIVED, THEN SEND THE         */
 3000 /*  NOTIFICATION TO THE USERNAME                              */
 3100
 3200            CHGVAR    VAR(&JOBNAME) VALUE(%SST(&MSGDTA 1 10))
 3300            CHGVAR    VAR(&USERNAME) VALUE(%SST(&MSGDTA 11 10))
 3400            CHGVAR    VAR(&JOBNBR) VALUE(%SST(&MSGDTA 21 6))
 3500            SNDMSG    MSG('YOUR JOB HAS BEEN INACTIVE FOR 20 +
 3600                        MINUTES, PLEASE SIGNOFF.') TOUSR(&USERNAME)
 3700            GOTO      CMDLBL(START)
 3800 ENDPGM
                           * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 12-1. Example of an Inactive Jobs Monitoring Program for QINACTITV/QINACTMSGQ.

QAUTOVRT Determines whether the system is to automatically create the necessary virtual devices or not. It has, however, no effect on PC Support. It is only valid for DSPT and Telnet.

QSECURITY Determines the level of security enforcement on the system. The system offers five levels of security, refer to Chapter 2, "System Values and Network Attributes" on page 2-1 for more information about system security levels. It is extremely important to use at least level 30 system security to protect the PC Support resources, and give you better possibilities to protect the folders from containing PC viruses. You can find a discussion about PC viruses in 12.6, "PC Viruses in AS/400 Folders" on page 12-20.

12.3 PC Support Conversations

PC Support applications consist of a PC ″requester″ and an AS/400 ″server″. In this way, each application requests a link to its AS/400 server. This link is an APPC conversation.

The program that controls the links between one or more of these conversations is called the *router*. The PC router also controls which systems the PC is connected to. The router has the APPC program necessary to communicate with AS/400, and starts different programs on the AS/400 system, depending on the PC request. Refer to "OS/2 Special Security Considerations" on page 12-3 for OS/2 special considerations.

Some PC Support functions start and stop the link automatically:

- Submit Remote Command
- Transfer function (interactive)
- Message Handling (send function)
- Data Queues Handling (send function)
- Remote SQL

Some PC Support functions maintain the link until they are explicitly stopped:

- Shared folders (for each drive assignment)
- Virtual Print (for each printer assignment)
- Work Station Function (for each display session)
- Messaging (receive function)

You should realize that a user is ″actively″ signed on to the AS/400 as long as at least one of the conversations is active. For each active conversation, you will have a job started on the AS/400. In this way, a PC Support user might have several AS/400 jobs active. When, for example, a user makes a transfer request from the AS/400 to the PC (RTOPC command), the PC router program evokes a matching AS/400 transfer program, and an AS/400 job gets started. When the file transfer finishes, the respective AS/400 job will finish too (the link is automatically started and stopped). Note that even though the users have finished all their active conversations, they are signed on to the system as long as the routers are active.

Every time the router is started, it asks for a user ID and a password. They are used to start the jobs on AS/400 and are stored in the PC′s memory. Every time a link is initiated, the router program sends both the user ID and the password to AS/400 for security verification. However, jobs that are already started will continue to use the user ID and password specified when they started.

The level of authority to the objects that the PC Support functions use is determined by the authority of the user who started the PC router. The user must be authorized to the objects. Note that this is independent of the user profile used in the interactive work station function job.

Another consideration is that the QINACTITV system value only affects the PC Support organizer and workstation functions. Other PC Support functions (for example, file transfer) are not affected by this system value. It is a security exposure if a user leaves a PC unattended with the PC Support router active.

12.3.1 Substitution of Passwords

In V3R1 the password is substituted before it is sent. APPC substitutes another character string, called a protected password, for a user password. The password is no longer sent in the clear on LANs or communication lines. This is a standard implemented in *all* APPC connections in V3R1.

When both ends support it, such as DSPT between two AS/400s using V3R1, the password is substituted. If one side lacks the support the password is sent in the clear as in earlier releases of OS/400.

The user cannot change this support (turn it on or off).

The user ID and password specified on an emulator′s sign-on screen does not get substituted when it is passed to the AS/400 from the PC. This is because the user ID and password are part of the data stream of the panel which does not use the password substitution technique. This is a security exposure as the sign-on panel datastream is visible in a communications trace. Using the ′bypass sign-on display′ option in Work Station Function and in CM 5250 emulation can prevent this at initial sign-on.

12.3.2 Handling PC Support Start Requests

PC Support requires a communications entry in the subsystem that is going to handle the communications. Generally, QCMN is the subsystem that handles the PC Support start requests. When the subsystem receives a PC Support start request, it locates a user profile for the job based on one of the following:

- The user ID and password sent (if they were specified)
- The user profile from the DFTUSR parameter of the subsystem′s communications entry

If a user ID and password are not sent when starting the router, the system checks the communication entry in the subsystem to see if it allows a default user. If the subsystem allows it, the start request is accepted. Specifying a default user in the subsystem′s communication entry (CMNE), is a security exposure. This gives the user the ability to start the PC Support without giving a user ID and password. To avoid this, a communication entry without a default user is setup in the subsystem description to handle the PC Support requests. An example of this communication entry, in comparison with a non-secure communications entry is:

Table 12-1. Subsystem Communications Entry for PC Support

Communications Entry	**Dev**	**Mode**	**Job Description**	**Default User**	**Max Active**
Non-secure CMNE	*ANY	*ANY	*USRPRF	QUSER	*NOMAX
Secure CMNE	*ANY	QPCSUPP	*USRPRF	*NONE	*NOMAX

Use the Display Subsystem Description (DSPSBSD) command to determine how the subsystem is defined. Observe the difference in Mode, and in Default User between the non-secure CMNE and the secure CMNE. This communications entry forces the user to supply a user ID and password to be able to access the AS/400. If there is a non-secure communications entry you should remove it.

If the subsystem assigns a default user profile, it uses it to establish the authorization to the objects and functions the job can reference. The subsystem also allocates a job description (which is specified in the subsystem′s communications entry). From here, the system handles the job as any other OS/400 job. If you want to know more about job management, refer to the *Programming: Work Management Guide*.

Every user who wants to use the PC Support functions must have a user profile. When creating a user profile, there are some parameters which will affect PC Support functions. The following table shows these security-related parameters and their effect on PC Support router initiated jobs (for example, file transfer, submit remote commands, remote SQL, data queues, shared folders, and so on).

Table 12-2. Security User Profile Parameters Related to Non-WSF PC Support Functions

User Profile Parameter	Affect PCS?	Comment
User Profile (USRPRF)	Yes	Used for STARTRTR.
Password (PASSWORD)	Yes	Used for STARTRTR.
User Class (USRCLS)	Yes	Special authorities are kept when using PC Support.
Special Authority (SPCAUT)	Yes	Special authorities are kept when using PC Support.
Limit Capabilities (LMTCPB)	Yes	The remote command function will be restricted. See note **1**
Group Profile (GRPPRF)	Yes	Like any other user profile.
Owner (OWNER)	Yes	Like any other user profile.
Group Authority (GRPAUT)	Yes	Like any other user profile.
Password Expiration Interval (PWDEXPITV)	Yes	See note **2**
Set Password to Expired (PWDEXP)	Yes	See note **2**
Limit Device Sessions (LMTDEVSSN)	No	You can start PC Support on more than one PC using the same user ID.
Job Description (JOBD)	Yes	You can specify a special job description for PC Support users, for example if you want them to have a specific library list.

Notes:

1 In V2R3 and earlier the limit capabilities (LMTCPB) parameter does not affect non-WSF PC Support functions. It could be a security exposure because a user, using the submit remote command function, can execute any CL command on the AS/400, even if he has limited capabilities specified in the user profile.

2 In V2R3 and earlier the password expiration control (*PWDEXPITV* and *PWDEXP*) do not affect the router′s password immediately. Even when the password expires on AS/400, users can still use the router without having to change their passwords. In this way, users can continue uploading or downloading files, for example, without any problems with expired passwords.

In V3R1 password validation is performed when the router is started. If the password has expired the user is forced to change it.

12.3.3 Common and Specific User IDs

During AS/400 PC Support installation, it is usually necessary to modify the PC Support configuration file (CONFIG.PCS) dependent on the physical connection and functions that the PCs will perform. An important parameter to consider is the user ID for the PC Support functions. PC Support allows two types of user IDs:

The *Common user ID* is the user ID used on all PC Support connections to any AS/400 system when a specific user ID is not supplied. If specified (in the RTCU entry in the CONFIG.PCS file), the user will be prompted for a valid password. If not specified, both user ID and password will be prompted for.

The *Specific user ID* is the user ID used for specific functional routers (local area network, SDLC, twinax, async, or communications manager if using OS/2). When specified, this user ID will be used for the STARTRTR. You can find more information about this in *PC Support/400: DOS Installation and Administration Guide* or *PC Support/400 OS/2 Installation and Administration Guide* if you are using OS/2.

If neither the Common user ID nor Specific user ID is configured, then the default user ID for the communications subsystem will be used.

To use shared folders function, PC Support users must be enrolled in the *System Distribution Directory*. You must have security administrator (*SECADM) special authority to enroll a user in the System Distribution Directory. Use the option 21 of the PCSTSK menu, or the command ADDDIRE, to achieve this task.

A security exposure can occur when issuing the STARTRTR (Start Router) command. It comes when you use the ″pipeline″ facility, which allows you specify a file name to automatically feed the user ID and password to the router. For example, you can specify:

```
                    File PIPELINE.TXT
                   ┌───────────┐
                   │  USER1    │
                   │  PASSWORD │
                   └───────────┘

          You can start the router in this way:

               STARTRTR < PIPELINE.TXT
```

Figure 12-2. Using Pipelining when Starting the Router

This is an obvious exposure and use of this technique should be discouraged. Not only will it allow automatic sign on to the router, this method is not guaranteed to work between releases, and the router interface may change.

12.3.4 Security Recommendations for PC Support Users

The following are recommendations for securing an AS/400 with PC Support users:

- Communications entry for mode for QPCSUPP should have *NONE for default user. If not change it using the CHGCMNE command.
- Each PC Support user should use a unique user ID.
- Avoid using COMMON USER IDs. Common user IDs are useful when you are trying to connect to many AS/400s using the same user ID and password from a single PC. They should not be used in an environment where multiple users use the same user profile and password to access router-based functions.
- Configuring user IDs for PC Support (CFGPCS):
  - If multiple users use the same PC:
    - Prompt for user ID and password
  - If the same person always uses the same PC:
    - Prompt for just the password
    - If connecting to multiple systems, use common user ID and prompt for the password in each system automatically
- Avoid using ″pipelining″ to feed user IDs and passwords.

12.4 PC Support Functions from a Security Point of View

In this topic, you can find information about some PC Support Functions and how they are affected by the AS/400 system security features.

12.4.1 Work Station Function

Work Station Function (WSF) gives the user access to the system in the same way as ordinary workstation users. There are no special security considerations. When you start the PC Support Work Station Function (in DOS) or 5250 Work Station Feature programs (in OS/2), an interactive job is started on the AS/400.

The system value QRMTSIGN allows you to define the system in a such way that a user can sign on to a WSF interactive job and bypass the AS/400 signon panel. The system will, however, evaluate whether the user who signed on the router job is authorized.

By changing the QRMTSIGN system value to *FRCSIGNON or *VERIFY the security will be enforced.

12.4.2 Transfer Functions

There is one big difference between an ordinary user and a PC Support user. The PC Support user has the ability to transfer data between the AS/400 system and the PC. Because of this, it is very important to secure the data in such a way that unauthorized access to the data by a PC Support user is avoided.

A security philosophy built on menu security might not be sufficient. You have to consider what level of authorities the users must have to the different objects they must access to get their job done. You need an authorization philosophy that gives each user the authorization they need, and no more.

To transfer data from the AS/400 to a PC the user must have at least *USE authority to physical files, and operational authority to logical files. To transfer data from a PC to AS/400, the user must have Object Operational, Object Management, Data Add, and Data Delete authorities. Refer to Table 4-3 on page 4-3 for more information on data authority types.

Note that when a file has been transferred from the AS/400 to the PC, security control is completely in the PC user′s hands.

It is only possible to do transfer requests when the PC router is active. However, an interactive job to the AS/400 does *not* have to be active. The users have to realize that an active PC router is equivalent to a signed on work station, and should be stopped before leaving the PC unattended. Special programs can be used to lock the PC keyboard. They require a user to supply the keyboard lock password in order to use the PC again. This feature is standard on OS/2.

12.4.3 Shared Folders

The *Shared Folders* function is used to store and access information, for example documents, PC programs, and PC files in folders on the AS/400. This gives the PC users a much larger disk storage capacity, the possibility to share the information between many PC users, and to share documents between OfficeVision/400 users with PCs or non-programmable terminals.

The PC Support shared folders function uses the standard AS/400 security options for folders. When you secure a folder (or an object inside a folder) you can specify the name of an authorization list, indicate whether it is personal, and give specific authority to users and group profiles.

Note that there is a big difference between libraries and folders. While libraries act as ″containers″ of objects, folders act like ″directories″. For example, if you specify *EXCLUDE for the public library authority, no user (except its owner, users with *ALLOBJ special authorities, or users authorized through an authorization list) will be able to access the library (they will get the message CPF9820: ″Not authorized to use the library XXXXX″).

If a user has authority to folder A and to an object within it, he or she is able to access that object. If folder A is a folder within folder B, the user can access the object in folder A even if he/she is excluded from folder B. Note that security for a folder is separate from security for objects in the folder.

Please observe that when you create an object in a folder it adopts the authorities of the folder.

OfficeVision/400 has a ″work on behalf of another user″ option. This option is not carried through to PC Support.

A convenient way to give several documents (or PC objects) or folders the same authorization is using authorization lists. Refer to Chapter 5, “Authorization Lists” on page 5-1 for more information.

Using the shared folders function, some PC commands can be used to perform tasks normally achieved by an AS/400 command. The PC command ″Make

Directory″ creates a shared folder on the AS/400 system, but the user must have the proper authority to the CRTFLR command on the AS/400.

If you want additional control over your shared folders, you should use exit programs. If you want more information about user exit programs, refer to 12.5.1, “PC Support Exit Programs” on page 12-15.

Properly implemented security of shared folders will enable you to protect them against PC viruses.

All PC Support users must be enrolled in the System Distribution Directory in order to use the AS/400 PC Support shared folder function. You must have security administrator (*SECADM) special authority to enroll a user to the System Distribution Directory.

Refer to Chapter 13, “OfficeVision/400” on page 13-1, for more information about shared folders.

12.4.4 Message Function

The PC Support *message function* allows PC users to send and receive messages to and from other users, regardless of whether the other users are PC users or workstation users. It is designed to be used by non-WSF users (only using router functions) to send and receive messages. WSF users can also use this facility, but this can lead to some confusion. Therefore, make the choice between PC Support message function, and the message function within the OfficeVision/400. Don′t use both.

Messages are handled with the PC Support message function when the PC router is active. An active PC router is therefore equivalent to a signed on work station, and should be stopped before leaving the PC.

If a user receives the messages on the PC, they are stored in a PC file. At this point, the messages are not protected by any AS/400 security functions. Security for these PC files is entirely the responsibility of the PC.

12.4.5 Submit Remote Command Function

With the Submit Remote Command PC Support function the PC Support user can submit AS/400 commands from the PC to be executed on the AS/400. An interactive Work Station display emulation need not be active. Because a PC Support user can enter commands in this way, we recommend that only those who need it are authorized to use the function. One way to do this is to just restrict authority to the RMTCMD.EXE file in the QIWS* folders. The other way is to restrict authority to the server program.

Another way to prevent from users issuing RMTCMD (Submit Remote Command) is to use an exit program, as discussed in 12.5.1, “PC Support Exit Programs” on page 12-15.

12.4.6 PC Support Organizer and Limited Capabilities Users

Menu security is a way of limiting access to data and commands in AS/400. Specifying an initial menu, an initial program, and limit capabilities *YES in the user profile restricts the user to the menu options. The user is prohibited from entering CL commands on the command line, specifying another menu, or another initial program on the signon screen.

When a user with limited capabilities starts the PC organizer (PCO) function, the PC program will work because the STRPCO command has ALWLMTUSR(*YES).

12.4.7 Summary of PC Support Conversations

- Some applications start and stop the link automatically:
  - Submit remote command
  - Transfer function (interactive)
  - Messaging (send function)
  - Data queues (send function)
  - Remote SQL
- Some applications maintain the link until they are stopped:
  - Shared folders (for each drive assignment)
  - Virtual print (for each display session)
  - WSF (for each display session)
  - Messaging (receive function)
- A user is ″actively″ signed on to the system as long as at least one conversation is active:
  - Each conversation is independently started and stopped. (Stopping a WSF session doesn′t affect shared folders)
  - To stop everything, use RMVPCS ALL or STOPRTR /F
- A user is signed on to the system as long as the router is active, even with no active conversations:
  - STARTRTR retains the user ID and password for starting future conversations
  - STOPRTR or RMVPCS must be used to stop new conversations from being started

12.5 Controlling Usage of PC Support Functions

There is a way you can control which activities a PC Support user is allowed to do. As mentioned in 2.8, “Network Attributes” on page 2-13, there are two parameters in your system′s Network Attributes which will help you to enhance the system′s security:

- **PCSACC** (PC Support Access) to control the following functions:
  - Virtual printer function
  - File transfer function
  - Shared folders Type 2 function
  - PC Support message function
  - Data queues function
  - Remote SQL function.

- **DDMACC** (Distributed Data Management Access) to control the following functions:
  - Shared folders Type 0 and 1 function
  - Submit Remote Command function

The default for these two parameters are *OBJAUT, which means that users are allowed to use the PC Support functions as far as their object authorities allow them. You can specify *REJECT if you do not allow any PC Support users at all, or you can specify the name of an exit program. Note that the DDMACC parameter is used for all DDM requests, not just requests from PC Support. It may be necessary to coordinate the setting of the DDMACC parameter with the person responsible for implementing DDM.

In V3R1, a new special value is added for the PCSACC network attribute called *REGFAC. This value allows you to use the new Registration Facility in V3R1 for registering exit programs for defined exit points. Exit points are automatically defined for all of the original PC Support servers. If you specify PCSACC(*REGFAC) in the network attributes, the server will look in the Registration Facility tables to determine which exit program to call. If no programs are defined, *OBJAUT is used. You must add exit programs to the exit points that you want to monitor using the Work with Registration Facility (WRKREGINF) command or the Add Exit Program (ADDEXITPGM) command. You can use the same program that you used for the PCSACC exit program, but now you specify exactly which servers you want to call the exit program. This support is not available for the DDMACC functions.

12.5.1 PC Support Exit Programs

PC Support/400 supports user-written exit programs. These programs enable you to enhance the system′s security. An exit program can determine whether the user has authority to call the appropriate PC Support function and have access to the data.

If an exit program is used, a data string is sent to the exit program when a PC Support function is called. The exit program can evaluate whether the user is authorized to the function and the requested objects.

Performance

If you specify a program to call for the PCSACC attribute, the program will be called for *all* of the exit points defined by the functions listed above. This can negatively impact your performance for all of these functions, not just the ones you are monitoring. If you are using OS/400 V3R1, we recommend using the *REGFAC value for the PCSACC attribute and so you can control which functions call the exit program.

The use of exit programs may affect your system′s performance because of the memory management to pass data, and the authorization check performed by the user exit programs.

The following topics show examples of exit programs to illustrate this important security facility.

Example of DDMACC Exit Program

Figure 12-3 has an example of an exit program that rejects the PC Support RMTCMD command. The program name must be specified in the DDMACC parameter of the network attributes to activate the program.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...
  100 PGM       PARM(&RTNCODE &DATA)
  200 DCL       VAR(&DATA) TYPE(*CHAR) LEN(30)
  300 DCL       VAR(&RTNCODE) TYPE(*CHAR) LEN(1)
  400 DCL       VAR(&FUNC) TYPE(*CHAR) LEN(10)
  500 CHGVAR    VAR(&FUNC) VALUE(%SST(&DATA 21 10))
  600             IF        COND(&FUNC = ′COMMAND   ′) THEN(CHGVAR +
  700                         VAR(&RTNCODE) VALUE(′0′))
  800 ELSE      CMD(CHGVAR VAR(&RTNCODE) VALUE(′1′))
  900 ENDPGM
                            * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 12-3. Example Exit Program to Reject PC Support RMTCMD Command. This will also work with SBMRMTCMD from another AS/400 user.

The program is called every time a user sends a request that involves the DDMACC parameter. The program sets the return code to ″0″ if the request is a PC Support RMTCMD request, and sets the return code to ′1′ for all other cases. The system accepts the return code ″1″ and rejects the return code ″0″.

Example of PCSACC Exit Program

This sample exit program is a bit more complex. It determines whether the user is allowed to perform file transfer and virtual print functions. The program uses two security files containing authorization information. Public authority for these files must be *EXCLUDE. As in the previous example, the system accepts the return code ″1″, and rejects the return code ″0″.

The data string that is passed to the exit program has different content, depending on the type of request. Refer to the *AS/400 PC Support DOS and OS/2 Technical Reference*, and to the *DDM User′s Guide*, for further details of exit program parameter fields, and for more examples of PC Support exit programs.

The Data Description Specifications (DDS) for the first security file, named PCSACCVP, are shown in Figure 12-4, and has only one field.

```
  100      A                                    UNIQUE
  200      A          R PCSACCVF
  300      A            USER         10
  400      A          K USER
```

Figure 12-4. Example Security File 1 for Virtual Print

The content of this field is the user who is allowed to use the virtual print function of PC Support. Figure 12-5 on page 12-17 has an example of the content of the first security file.

```
        USER ID:

        FRED
        MARTIN
        OERJAN
        PIA
        STELLA
```

Figure 12-5. Example Content of Security File 1 for Virtual Print

The second security file (Figure 12-6), in this example named PCSACCTP, contains the users who are allowed to use the file transfer function, and the requests they are allowed to do on selected libraries (you could also have specified which files the users are allowed to access).

```
     200      A                                    UNIQUE
     300      A          R PCSACCF
     400      A            USER          10
     500      A            APPLIC        10
     600      A            RQST          10
     700      A            LIB           10
     800      A          K USER
     900      A          K APPLIC
    1000      A          K RQST
    1100      A          K LIB
```

Figure 12-6. Example Security File 2 for Transfer Requests

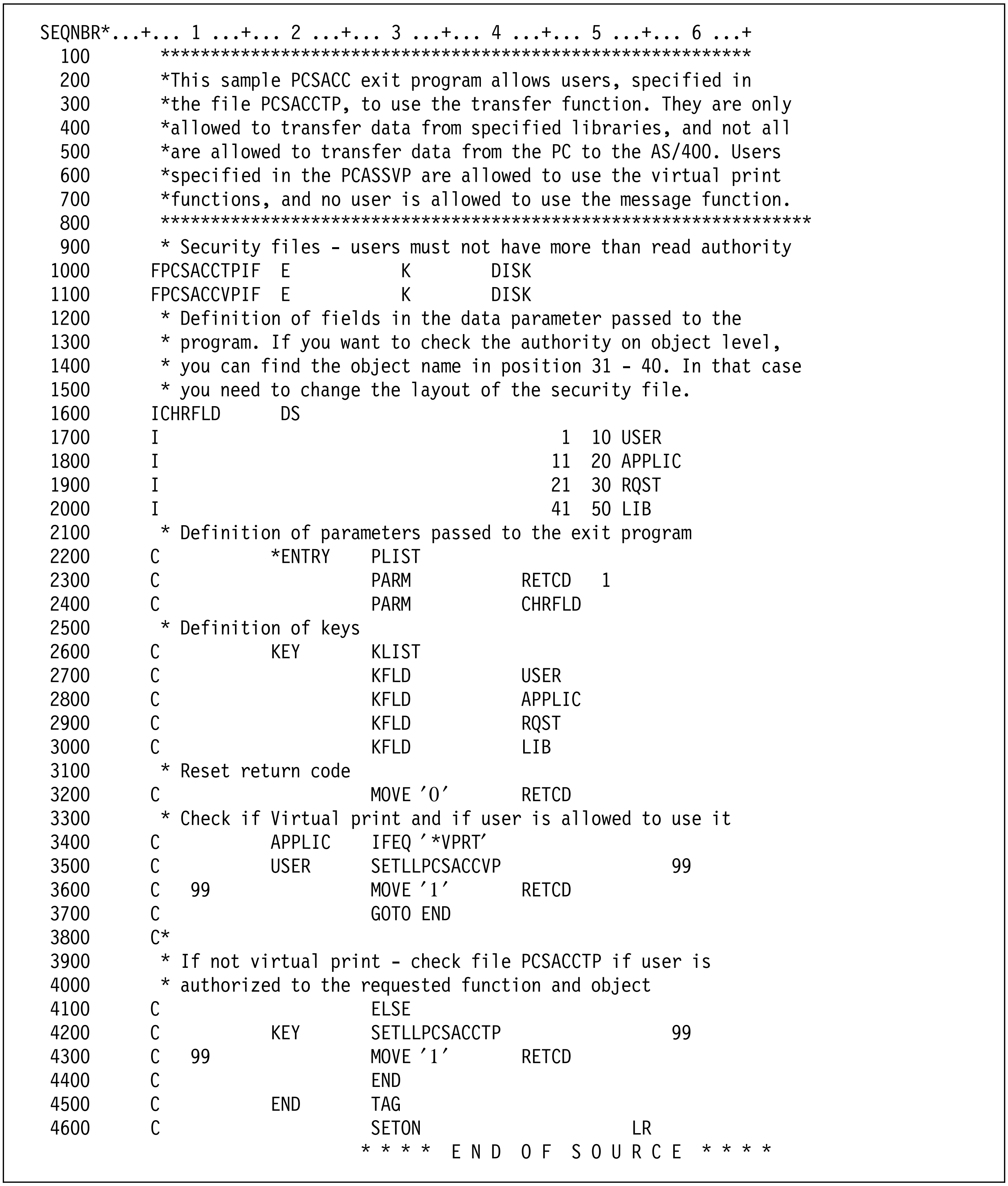
You can find an example of the content of the second security file, used for file transfer functions in Figure 12-7.

```
USER:       APPLIC:     RQST:       LIB:

MARTIN      *TFRFCL     EXTRACT     SECURITY
MARTIN      *TFRFCL     SELECT      SECURITY
OERJAN      *TFRFCL     EXTRACT     SECURITY
OERJAN      *TFRFCL     JOIN        SECURITY
OERJAN      *TFRFCL     REPLACE     SECURITY
OERJAN      *TFRFCL     SELECT      SECURITY
PIA         *TFRFCL     EXTRACT     SECURITY
PIA         *TFRFCL     SELECT      SECURITY
STELLA      *TFRFCL     EXTRACT     SECURITY
STELLA      *TFRFCL     JOIN        SECURITY
STELLA      *TFRFCL     SELECT      SECURITY
```

Figure 12-7. Example Content of Security File 2 for Transfer Requests

The example RPG program (Figure 12-8 on page 12-18) allows specified users to use file transfer to the extent that the transfer security file (in this example called PCSACCTP) allows, and to use virtual print, when they are defined in the security file (in this example called PCSACCVP).

Refer to the *AS/400 PC Support/400: DOS and OS/2 Technical Reference*, and to the *AS/400 DDM User's Guide*, for further details of exit program parameter fields, and for more examples of PC Support exit programs.



```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+
  100       ************************************************************
  200       *This sample PCSACC exit program allows users, specified in
  300       *the file PCSACCTP, to use the transfer function. They are only
  400       *allowed to transfer data from specified libraries, and not all
  500       *are allowed to transfer data from the PC to the AS/400. Users
  600       *specified in the PCASSVP are allowed to use the virtual print
  700       *functions, and no user is allowed to use the message function.
  800       ******************************************************************
  900       * Security files - users must not have more than read authority
 1000      FPCSACCTPIF  E           K        DISK
 1100      FPCSACCVPIF  E           K        DISK
 1200       * Definition of fields in the data parameter passed to the
 1300       * program. If you want to check the authority on object level,
 1400       * you can find the object name in position 31 - 40. In that case
 1500       * you need to change the layout of the security file.
 1600      ICHRFLD      DS
 1700      I                                        1  10 USER
 1800      I                                       11  20 APPLIC
 1900      I                                       21  30 RQST
 2000      I                                       41  50 LIB
 2100       * Definition of parameters passed to the exit program
 2200      C           *ENTRY    PLIST
 2300      C                     PARM           RETCD   1
 2400      C                     PARM           CHRFLD
 2500       * Definition of keys
 2600      C           KEY       KLIST
 2700      C                     KFLD           USER
 2800      C                     KFLD           APPLIC
 2900      C                     KFLD           RQST
 3000      C                     KFLD           LIB
 3100       * Reset return code
 3200      C                     MOVE ’0’       RETCD
 3300       * Check if Virtual print and if user is allowed to use it
 3400      C           APPLIC    IFEQ ’*VPRT’
 3500      C           USER      SETLLPCSACCVP                 99
 3600      C   99                MOVE ’1’       RETCD
 3700      C                     GOTO END
 3800      C*
 3900       * If not virtual print - check file PCSACCTP if user is
 4000       * authorized to the requested function and object
 4100      C                     ELSE
 4200      C           KEY       SETLLPCSACCTP                 99
 4300      C   99                MOVE ’1’       RETCD
 4400      C                     END
 4500      C           END       TAG
 4600      C                     SETON                     LR
                                * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 12-8. Example PCSACC Exit Program

12.5.2 PC Support Server Programs

There is another way you can restrict the usage of PC Support functions. In Table 12-3 on page 12-19 you can find a list of PC Support server programs and their functions.

Table 12-3. PC Support Server Programs

Program	**Library**	**Description**
QCNPCSUP	QSYS	Authority Checking
QXFINIT	QIWS	Shared folders Type 2
QPWSSTPO	QSYS	V3R1 File server (original DOS, DOS extended and OS/2 clients; Shared folders Type 2)
QCNTEDDM	QSYS	DDM (Shared Folders Type 0 and 1, and Remote Cmd)
QTFDWNLD	QIWS	Transfer Facility
QVPPRINT	QIWS	Virtual Printer
QMFRCVR	QIWS	Messaging Receiver
QMFSNDR	QIWS	Messaging Sender
QRQSRVX	QIWS	Remote SQL
QHQSRV0	QIWS	Remote SQL
QHQSRV1	QIWS	Remote SQL

If you secure any of these programs you will be securing the respective PC Support function. For example, if you secure QCNTEDDM program setting its public authority to *EXCLUDE, no PC Support user without private authority will be able to execute the remote command function, and will get the PC message: `5056 Security error occurred for system X`. Additionally, a message will be displayed at the QSYSOPR message queue (or at the QSYSMSG message queue, if you are using it) saying: `Program start request received on communications device Y was rejected with reason codes 709, 1506`. This offers better performance that the exit program approach, but there is no guarantee of future support.

Note: It may be useful to start with these programs, before making an exit program, if you are not quite sure who uses PC Support, and how. When a person yells because he/she is excluded from a function needed, you know a little more about what the future exit program must cover. It takes seconds to authorize the excluded person.

12.5.3 Security in Cooperative Environments

Cooperative processing between the PC (running under DOS or OS/2) and the host (AS/400 in our case) has become a fundamental strategy in the world of computers.

The fundamental concept in cooperative processing is to have a PWS (Programmable Work Station) user who, signing on only once, will be able to access a variety of applications, maybe running on different systems. With cooperative processing, the application program performing the different tasks may be divided between two (or more) systems to take advantage of their individual and combined strength.

In this sense, PC Support/400 has a set of programs to let the users simultaneously use facilities of a PC and have access to AS/400 resources. For cooperative processing applications, you can take advantage of the different facilities in:

- Shared folders

- Virtual Printer
- Message function
- Transfer Function
- Remote SQL
- Workstation function
- AS/400 data queue support
- Remote command support

Actually, the router consists of three elements:

- Resident element
- Starting and ending elements
- Transaction programs

The *resident element* performs the different LU6.2 functions and provides an application program interface (API) for conversation requests.

The *starting and ending elements* let the router establish and control connections with the different systems.

The *transaction programs* perform user application processing in cooperation with their counterparts, such as APPC programs in AS/400.

The router has a special characteristic. It supports user verification. This allows the host system to verify the identity of the user who is starting a request before granting access to the transaction program and its resources. The user ID and password are entered by the user when the connection is made to the remote system. If security values are incorrect, or if the user is not authorized to the resources he requests, the host is able to deny the access.

If you want more information about cooperative processing on AS/400, refer to the *PC Support/400: DOS and OS/2 Technical Reference*, *PC Support/400: Application Program Interface Reference* and to the Redbook *Cooperative Processing and Graphical User Interfaces in an AS/400 Environment*.

12.6 PC Viruses in AS/400 Folders

A PC virus is a program running in a PC. When stored in shared folders they can spread to other PCs when they are accessed. Although these viruses can be very damaging to an attached PC, there are no known PC viruses that damage AS/400 programs or destroy the integrity of AS/400 databases.

12.6.1 Detecting a PC Virus

To detect a virus, you must periodically run a virus scan program against the AS/400 folders that contain executable PC programs. This can be done from any PC that has PC Support installed, and a system drive assigned to shared folders.

IBM sells a virus-detecting program called the *IBM Anti-Virus Product*. Your IBM marketing representative can help you get this program.

The possible risk of introducing a virus with the virus scan program must be weighed against the possibly greater exposure of not performing the scan. This risk can be minimized by either storing the virus scan program on a write-protected diskette and start the PC from that diskette when you run the program, or by using a user profile which only has read access to the folders being scanned. It can be a user profile to be used just for scanning viruses.

Another detection technique is to use the audit journal to detect authority failures. If the user running the program only has read authority to the objects in the folders, and an authority failure occurs, you know that the program tries to update something it shouldn′t. Authority failures are logged when the audit level is set to *AUTFAIL. The authority failures can be selected from the journal receiver by requesting entries of type AUTHORITY FAILURE (AF). Audit journals are explained in Chapter 6, “Audit Journal” on page 6-1.

12.6.2 Preventing a Virus

Set your system security level to at least 30 so that the system′s resource security can be used to help prevent PC viruses from entering a folder. Without at least a level 30 security on your system, all users automatically have system administrator level authority, and cannot be restricted to read-only access of objects.

Secure the shared folders that contain PC programs so that these folders are read-only (*USE). This probably means reorganizing the shared folders so that executable programs (.COM, .DLL, .EXE, .OVL, and so forth) are in different folders from PC data files and ordinary documents.

Securing the folders with read-only authority will not prevent users with system administrator (*ALLOBJ) special authority to store PC programs in the folders. The audit journal will not report ACCESS DENIED messages for these users.

When executable program updates are required, use a user profile specifically reserved for this purpose. A virus scan program should be run immediately before and after an update.

Using Exit Programs

Another prevention technique is to use an exit program, such as the NOVIRUS.RPG in the QIWSTOOL shared folder. This folder is shipped with PC Support. NOVIRUS.RPG is provided as a sample of an exit program that can be changed to fit your requirements. There is a listing of the program in Figure 12-9 on page 12-22. If you specify an exit program (via the CHGNETA command’s PCSACC and DDMACC parameters), it is run by the system whenever a document (PC file or program) in a shared folder is opened. This exit program can be used to prevent attempts to write data into any document (PC program) in a shared folder. The following files are checked: *.COM, *.DLL, *.EXE, *.OVL.

Using exit programs may be useful to prevent the unintentional storing of a program that has been modified. It can also be used to prevent the user profile used to run the virus detecting program from doing any updates at all.

```
 **************************************************************************
 *                                                                        *
 * PROGRAM : PCVIRUS -- This program can be specified as an exit          *
 *           program on the CHGNETA parms DDMACC or PCSACC to             *
 *           prevent attempts to write data into any PC program           *
 *           stored on the shared folders.                                *
 * PURPOSE : This may be useful to prevent the unintentional              *
 *           storing of a program that has been modified.                 *
 * Restriction : This program must not be active when installing          *
 *               a new release of the system.                             *
 *                                                                        *
 **************************************************************************
IPAA         DS
I                                        1  10 PUSR
I                                       11  20 PAPPL
I                                       21  30 PRQS
I                                       31  40 POBJ
I                                       41  50 POBJL
I                                       51  60 PMBR
I                                       61  70 PFMT
I                                       71  750PLEN
I                                       76  85 PLU
I                                       86  95 PSRV
I                                       96  96 PREAD
I                                       97  97 PWRITE
I                                       98  98 PUPDAT
I                                       99  99 PDLT
I                                      100 112 PCOBJ
I                                      101 112 PCOBJ2
I                                      113 176 PCDIR
C           *ENTRY    PLIST
C                     PARM           RTNCD   1
C                     PARM           PAA
C**************************************************************************
C*                                                                        *
C* If the operation is open for write or update to a file that is         *
C* named with an .EXE  .COM  .OVL or .DLL  then reject the request.       *
C*                                                                        *
C* If request is 'CREATE' SCAN name starting in position 2 for a          *
C* file extender that corresponds to any of .COM  .EXE  .OVL  or .DLL     *
C*           ELSE                                                         *
C*    IF request is open or update, SCAN the name starting in position 2  *
C*    If found reject the operation (rtncd = '0')                         *
C*      ELSE operation is OK        (rtncd = '1')                         *
C*                                                                        *
C**************************************************************************
C                     MOVE '1'       RTNCD
C*          PUSR      IFEQ 'NIK'                      TEMP                *
C           PRQS      IFEQ 'CREATE'
C           '.EXE'    SCAN PCOBJ2                   10PC FILE OF EXE
C  N10      '.COM'    SCAN PCOBJ2                   10PC FILE OF COM
C  N10      '.OVL'    SCAN PCOBJ2                   10PC FILE OF OVL
C  N10      '.DLL'    SCAN PCOBJ2                   10PC FILE OF DLL
C                     ELSE
C           PRQS      IFEQ 'OPEN'
C           PWRITE    IFEQ '1'                        OUTPUT REQUEST
C           PUPDAT    OREQ '1'                        UPDATE REQUEST
C           '.EXE'    SCAN PCOBJ2                   10PC FILE OF EXE
C  N10      '.COM'    SCAN PCOBJ2                   10PC FILE OF COM
C  N10      '.OVL'    SCAN PCOBJ2                   10PC FILE OF OVL
C  N10      '.DLL'    SCAN PCOBJ2                   10PC FILE OF DLL
C                     END
C                     END
C                     END
C   10                MOVE '0'       RTNCD
C*                    END                             TEMP                *
C                     SETON                         LR
C                     RETRN
                     * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 12-9. Program to Check for PC Program Modifications

12.7 Detecting Modifications of PC Programs

The previous topic described how object security and user exit programs may help to prevent the introduction of a “known” virus to PC programs. This topic takes you one step further by describing how you can detect *which* PC programs may have been infected by an “unknown” virus by using the object change date. Although virus scanning programs are useful for finding infected programs for a ″known″ virus (because they look for known bit patterns to identify a ″known″ virus), they may not detect an “unknown” or new virus.

Detecting which PC programs may have been infected by an ″unknown″ virus is a three step process: obtain the last-changed-date, obtain the external name, and produce a report.

12.7.1 Using the Last-Changed-Date

PC programs are stored using the OS/400 object type of *DOC. Documents (like other OS/400 objects) have a last-changed-date that can be used to detect *when* an object was last modified. The last-changed-date for an object is automatically updated when the object is modified, and cannot be reset. Documents also have a revision date that changes less frequently, but this date is not secure and can be reset to its previous value by a PC virus.

12.7.2 Obtaining the Last-Changed-Date

You can store the last-changed-date for all *DOC objects in a database file using the OUTFILE option of the Display Object Description (DSPOBJD) command. Use the following command to create a database file named OBJD for all documents stored in the system:

```
DSPOBJD   OBJ(QDOC/*ALL)
          OBJTYPE(*DOC)
          OUTPUT(*OUTFILE)
          OUTFILE(OBJD)
```

The object name from the DSPOBJD command is a system-generated name. This system object name is not meaningful to most users and therefore you need to obtain the external name (document name).

12.7.3 Obtaining the External Name

Both the system object name and the user-assigned name can be written to a database file using the Query Document Library (QRYDOCLIB) command. Use the following command to create a database file named QRY for all documents in all folders:

```
QRYDOCLIB  FLR(*ALL)
           OUTFILE(QRY)
```

Depending upon the number of documents in the system, the DSPOBJD and QRYDOCLIB command may take a long time to run, and should be submitted as a batch job. To ensure access to all documents, the batch job should be run by an *ALLOBJ user.

12.7.4 Producing a Report

After the two database files OBJD and QRY are produced, the database support can be used to join the two files using the system object name. The following SQL/400 statements produce a report that provides the change date, and both the system and user-assigned names.

```
SELECT   objd.odldat, qry.qdlonm,
         qry.qdldnm, qry.qdlflr
FROM     objd, qry
WHERE    objd.odobnm = qry.qdlonm
ORDER BY qry.qdlflr
```

Note that you can get the same results using the Query/400 licensed program. The field names from the above example can be used in other programs to produce a similar report. If you select changes for specific dates, you can also produce a subset of this report as shown below:

```
  Change  OBJECT      DOCUMENT      FOLDER
  Date    NAME        NAME          NAME
 -----------------------------------------------
  071891  EJKN343486  NEWDOC        EVANS
  070891  EJKN551026  WOEPCS.BAT    EVANS
  070891  EJKN561146  STARTRTR.EXE  EVANS
  070891  EJST142650  WOEPCS.CPY    EVANS
  070991  EJLL411130  QHSTPRT       QFOS2950
  070991  EJLL411306  QINDUSR       QFOS2950
  070991  EJLL411440  QPROFDOC      QFOS2950
  070991  EJLL411564  QPROFNOT      QFOS2950
 ********  End of data  ********
```

Review the change date of your objects to see which PC programs possibly may have been infected by an unknown virus.

12.8 How PC Viruses Can Spread

As mentioned earlier PC viruses can spread from a folder in AS/400 to PCs using PC Support. Following are some of the most common ways PC viruses are spread:

- Copying/Sharing programs:
  - Demo distributions
  - Informal, often illegal, copies of a product
  - Shared utilities via diskette
- Sharing workstations/PCs:
  - Open access to PCs (for example, shared personal and business use)
  - Exchange/Load of hardware
  - Demonstration centers
  - Training machines
  - Repair centers
- Transporting programs:
  - Hand carrying (demos, utilities, for example) between office and home/customers
  - Mailing demo or update diskettes
  - Download from Bulletin boards or uncontrolled sources

12.8.1 Reducing the Risk of PC Viruses

Try to avoid PC viruses. They can cause considerable damages and expenses. Here are some precautions:

- Follow good security practices:
  - Educate the users about security threats, including PC viruses
  - Secure unattended PCs
  - Keep good backups of all important data
- Reduce the possibility of being infected:
  - Ensure adequate controls exist on software libraries, development systems, and other similar areas
  - Check for viruses in new software
  - Limit the possibility to create or install new programs
  - Shut off idle PCs, or lock their keyboard
  - Isolate from sources of infection, such as networks and programs from the ″outside″
- Take steps to ensure that virus infections will be detected quickly:
  - Educate users about possible warning signs
  - Use programs that detects if unintended modifications of programs and data occur
  - Make sure the users know how to report a potential problem
- Take steps to deal with virus infections when detected:
  - Develop a plan to deal with viruses before an infection occurs
  - Isolate infected systems until they are cleaned up
- Be prepared to recover from a virus after it has been contained:
  - Be able to restore programs and data from virus-free backups
  - Maintain a careful watch for re-infection

12.9 Recommendations Summary

The control of the functions in PC Support involves many tasks:

- A PC with an active PC router, even with no signed on WSF user, is to be viewed as an *active workstation*. It is possible for anyone to use the message function, to use the Submit Remote Command function, and to access data in the AS/400 system, to the extent of the authorization of the user ID used to start the PC router. Ensure that PC Support users finish their work with both signoff from the interactive WSF session, and by ending the router. Before ending the router, you should end any active PC Support function (you have PC commands to do it) and then issue the STOPRTR command (both in DOS or OS/2). If you are working with OS/2, once you finish the router, stop the 5250 communications feature of Communications Manager.
- Limit the system values QAUTOVRT and QMAXSIGN in a manner that limits the number of invalid sign on attempts from any user.
- Add a new communications entry to the subsystem that handles the router job as shown in Table 12-1 on page 12-8 to force a user to supply a user ID and password. Don′t use the default communications entry.

- Limit the user access to submit remote command, by either revoking public authority from the command, or by using an exit program and change the DDMACC parameter of the AS/400 network attributes.
- Actively scan shared folders for viruses. Don′t wait until you have a problem to try to solve it. You should run your viruses scanner frequently (for example once a month).
- If you are using DOS, avoid the automatic signon capability to start the PC Support workstation function. It could be a security exposure, but if you use it, always issue the stop workstation function command (STOPWSF) on the PC after you sign off from AS/400.

12.9.1 Keyboard Lock

When a PC Support user signs off from the AS/400 session, the router still remains active until the STOPRTR command is issued. Any person may use that PC (if left unattended) and perform all the PC Support tasks available to the user who started the router.

A keyboard lock, implemented in either the PC hardware or software is strongly recommended.

A PC program that monitors the AS/400 session using the WSF API could be written to stop the router, with a “force all to stop” (STOPRTR /F) whenever a user signs off from the AS/400 session. However, this require the user to restart the router. All PC Support services that were being used will also have to be restarted.

Chapter 13. OfficeVision/400

OfficeVision/400 is a licensed program that operates on the AS/400 allowing users to maintain folders, documents, and calendars, and to exchange messages, notes, and documents, (hereafter called *distributions*), with users on the local AS/400 as well as with users on remote systems.

This chapter will discuss security for OfficeVision/400 and possible exposures when exchanging information with other systems.

When exchanging information with other systems, OfficeVision/400 makes use of the concepts discussed in Chapter 10, “Communications Security in SNA” on page 10-1 and in 11.9, “SNA Distribution Services (SNADS)” on page 11-16. This chapter is concerned only with OfficeVision/400 application, not the communications part. OfficeVision/400 involves both sending and receiving distributions either on the same system or between different systems. When establishing security in this environment both situations should be considered together.

This chapter will be divided into the following sections:

- Overview of Document Library Services Security
- Overview of OfficeVision/400 Security
- Standalone OfficeVision/400
- Exchanging distributions with remote systems

Each section will contain a description of the environment and the functions. Each section will end with a list of considerations that the Office administrator should make when setting up Office.

In this chapter the word “user” means Office user, unless otherwise stated.

13.1 Overview of Document Library Services Security

This section discusses general considerations for security in document library services. Document library services security is not a separate scheme on an AS/400 and must be planned together with security considerations for the whole system.

We will first look at some terms and definitions for Document Library Services, and then what is covered by Document Library Services.

13.1.1 Terms and Definitions for Document Library Services

The following are important concepts with regard to document library services:

Access Codes
A four-digit code used to:

- Group together documents and folders
- Control access to documents and folders

Folder A directory for documents. A folder is used to group related documents and to find documents by name.

Documents

A document is any collection of data stored in a document object. All documents and folders on a single AS/400 system make up the document library. A document can contain any type of data stored in it by an application. For example, OfficeVision/400 can store notes, memos, reports, and other items; the PC Support/400 shared folders application can store any data that could otherwise be stored in a PC; an AS/400 application can store any data into a document by using CL commands, such as FILDOC and RPLDOC.

Document Library Objects

Folders and documents.

13.2 Document Library Services Security

The document library object security is based on system object security. There are extensions to system security functions that apply to documents, folders, and mail. The sequence of checks used to determine a user′s authority to a document library object is described in the *Planning for and Setting Up OfficeVision/400* manual.

A user must be enrolled in the system distribution directory to access a folder or a document.

Securing Folders and Documents

Securing a folder secures:

- Folder table of contents
- Folder description
- Folder management

Securing a document secures:

- Document content
- Document description
- Document management

The security concepts that apply to the document library are:

- Object authority
  - Owner
  - Private authorizations
  - Public authorizations
- Additional sources of authority
  - Special authorities
    - *ALLOBJ
    - *SECADM
    - *SAVSYS
  - Authorization lists
  - Group profiles

Creating New Objects

The default authorities to newly created objects are:

- When created in a folder the authorizations are the same as the folder they are created into.
- When folderless (via FILDOC, CPYDOC, QRYDOCLIB, CRTFLR) the authorizations are set to system defaults of:

  User creating is the owner
  Public is *EXCLUDE
  Sensitivity is I=NONE
  No explicit authorities
  No access codes
  No authorization lists

Work on Behalf of Another User

When working on behalf of another user:

- The system thinks you are the other user
- Applies to folders, documents, and mail
- Authorities to commands are not assumed
- Automatically excluded from folders and documents marked as ″personal″

Group Profiles

The following applies to group profiles:

- When a folder or document is created the user who creates it becomes the owner. The OWNER parameter in the user profile has no effect (defaults to *USRPRF).
- The CHGDLOOWN command can be used to transfer the ownership to a group profile, provided the group profile is enrolled in the system directory.
- No authority is automatically given to the group profile. The GRPAUT parameter in the user profile has no effect (defaults to *NONE). The ADDDLOAUT or EDTDLOAUT commands can be used to give explicit authority to a group profile, provided that the group profile is enrolled in the system directory.

Only the owner, or someone with *ALLOBJ or *SECADM special authorities, are able to change the ownership of an object.

13.2.1 Creating Folders

Certain security controls need to be in place to restrict users from creating folders on the system such that the number of folders is controlled.

The *ROOT folder

The *ROOT folder contains all the folders and documents within the document library. It includes both the first level folders and the folderless documents.

The *ROOT folder′s public authority can be used to restrict or control the creation of first-level folders on the system. It does not, however, prevent a user from creating as many folders as they wish within their first-level folder. If the public authority is changed from *CHANGE (the default) to *USE, only users with *ALLOBJ or *SECADM special authorities can create first-level folders.

Creating First-Level Folders

It is possible to prevent a user from creating new folders by limiting their access to the CRTFLR command. If a user is excluded from the command, he or she will not be able to create new folders, but can still create, copy, change, read, or delete documents in existing folders, depending on his authorities to the folders and the documents.

The CRTFLR command is the only command where excluding the user has an effect inside Office. Therefore, excluding the user′s authority to the command outside Office will have no effect inside Office.

13.2.2 Object Ownership

Users can belong to a group profile, and optionally all objects created by that user can be owned by the group profile. (To do this, specify *GRPPRF in the *OWNER parameter of the CRTUSRPRF command). In Office, however, a folder or document is owned by the user profile that created it. Ownership can be transferred to a group profile using the CHGDLOOWN command, or through an administrator′s menu. The group profile must be enrolled in the system directory.

When a user is enrolled in Office, the user will become the owner of folders and calendars created at that time. Public authority will automatically be *EXCLUDE.

If the folders and calendars were created before enrollment of the user, the user who created them will be the owner. The user who creates documents in the folder will be the owner of those documents.

The owner of folders and documents is responsible for granting *ALL authority to the user responsible for the save procedures, unless this person is granted *SECADM or *SAVSYS special authority. Implications of granting *SECADM authority will be discussed in the section 13.2.5, “Authority” on page 13-8.

Recommendations for Object Ownership

- Use authorization lists to grant authority to documents and folders for save purposes, unless the user who perform the save has *SAVSYS special authorities.

13.2.3 Authorization Lists

Normal AS/400 security applies to Office. Accordingly, authorization lists should be used for a flexible and easy-to-use method for securing documents.

Securing documents on an object authorization basis would require changing access to each document, when a new need-to-know situation arises. Changing an authorization list means changing access for all documents secured by this list. Object authorization can still be used to expand or limit authorization list securing. See Table 5-1 on page 5-11 for a comparison of group profiles and authorization list features.

An average user with an initial menu in Office will not have authority to create authorization lists. If the Office administrator creates and maintains authorization lists, he will have *ALL authority to all documents and folders secured by the list. Users can be allowed to create lists on the option 50 on the Office main menu if the structure of the Office application is that all users maintain their own authorization lists.

Authorization is further discussed in the section 13.4.1, “Create Folders” on page 13-23.

13.2.4 Access to Document Library Objects

A user having *ALLOBJ special authority has access to *all* folders and documents, even if not enrolled in Office. As a user with *ALLOBJ special authority has access to almost everything on an AS/400, it is impossible to exclude such a user from access to Document Library Objects. All other users on the system can be specifically excluded from a given Document Library Object.

Users may be allowed to create their own authorization lists. The following example illustrates the user capabilities:

- USER1 is included on an authorization list created by USER2.
- He has the authority *ALL to documents secured by this authorization list.
- USER1 has been granted *SPCAUT(*NONE) on his user profile.

  1. If USER1 is enrolled in Office he or she can change, copy, read, and print the documents secured by this list. He has the options both inside Office, using the Office menus, and outside Office, using commands. Outside Office USER1 must have access to a command line and be authorized to use the commands.
  2. If USER1 is not enrolled in Office, but enrolled in the System Distribution Directory, he can use some *FLR and *DOC commands to work on the documents.
  3. If USER1 is not enrolled in Office and not enrolled in the System Distribution Directory, he will not be allowed to use *FLR and *DOC commands from a command line outside Office.

When establishing a structure for Office, you may consider allowing the System Security Officer to be responsible for maintaining authorization lists. This way only one user profile will be the owner of several objects used in Office. Users on the authorization list will have the same authorities as they would have if the authorization list was created by another user, granting them authority to his documents and folders. The users that create new documents and secure them by the authorization list, created by *SECOFR, will still be the owners of the documents and have *ALL authority to them. *Planning for and Setting Up OfficeVision/400* includes a discussion on areas of work for a *SECADM.

The general rule that applies to Office is that no user can get into Office unless the user is enrolled by a user with *SECADM authority. The only exception from that rule is the user ID of QSECOFR that is automatically enrolled and has the ability to enroll himself if he should be deleted. All other user IDs must be enrolled by either QSECOFR or a user with *SECADM authority.

It is possible to use some of the Office functions outside Office by executing the command from a command line. Some rules apply to these commands. The user has to be enrolled in Office to use the commands even outside Office. The authorities defined for the Document Library Objects inside Office applies for the commands executed outside Office too. Table 13-1 on page 13-6 shows the possibility to view a document outside Office by the command DSPDOC. The command will display the contents of the document if authorization rules allow it, as illustrated in Figure 13-1 on page 13-7.

USER	Enrolled in Office	*ALLOBJ	Result of command
*ANY	YES	YES	Document displayed
		NO	Document displayed, dependent on authority to document in Office
	NO	YES	Msg: User not enrolled in Office
		NO	Msg: User not enrolled in Office
*SECOFR	YES	YES	Document displayed
	NO	YES	Msg: User not enrolled in Office

Table 13-1. The Result of Using the Command DSPDOC. The heading “Enrolled in Office” means enrolled in Office and System Distribution Directory for this table

A user not enrolled in Office and the System Distribution Directory cannot get to the Document Library Objects outside Office, as the system checks the user′s enrollment before the command is executed.

A user enrolled in the System Distribution Directory, but not in Office, will have access to certain commands regarding Document Library Objects. One of the commands is WRKFLR, which will list all folders that the user is authorized to, and with *ALLOBJ authority the user will have access to all folders on the system. From this display a user will be able to perform any option listed on the screen.

These options are valid only if the user is enrolled in the System Distribution Directory.

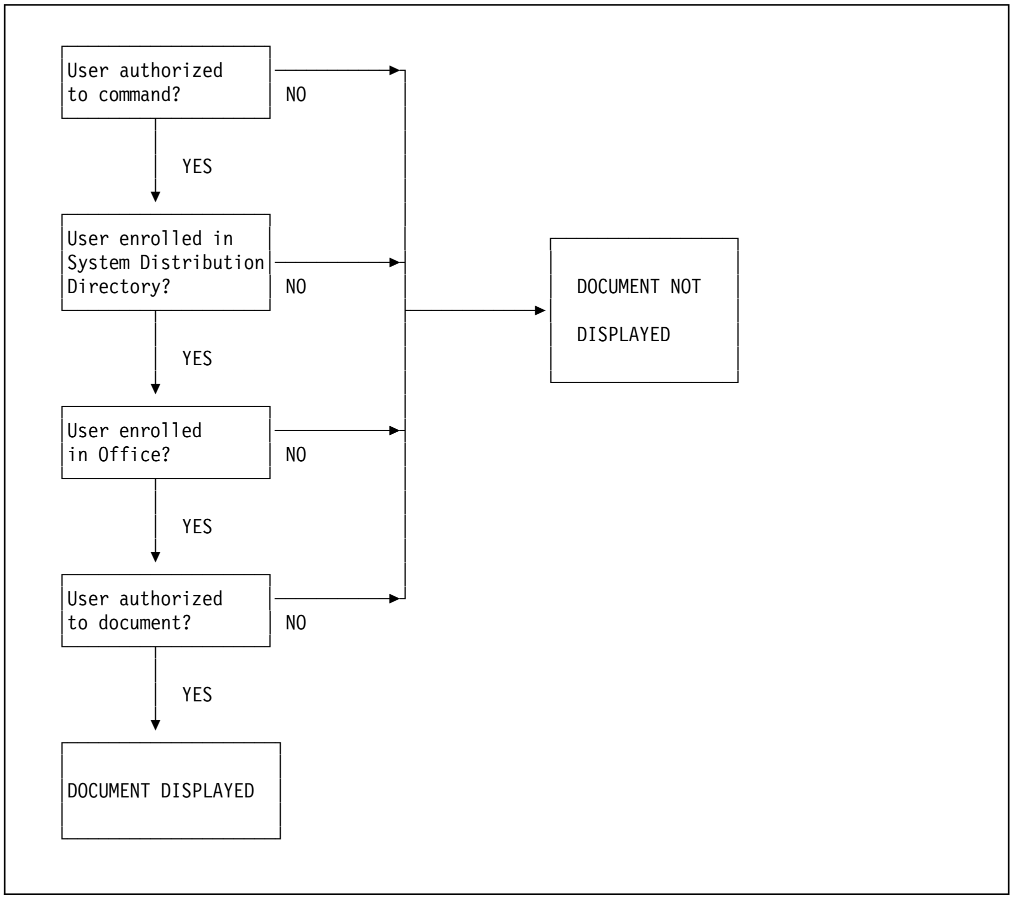

User authorized to command?
NO
YES
User enrolled in System Distribution Directory?
NO
YES
User enrolled in Office?
NO
YES
User authorized to document?
NO
YES
DOCUMENT DISPLAYED
DOCUMENT NOT DISPLAYED

Figure 13-1. Authority Testing for Command DSPDOC. Authority is tested first for the command, then for enrollment, and then for authority to the document.

Working with commands such as WRKAUTL, WRKOBJOWN, or WRKOBJ, a non-Office user can get to the system-assigned object names of the document library objects. The ′Work with Authorization Lists′ display has an option for displaying which objects are secured by this list.

The Work with Objects by Owner screen will display the names and the type for all objects owned by a user. For document library objects, this may be either the user-defined name, if there is one, or the system-assigned name.

The Work with Objects display will display all objects in a given library. It will show all the system names for the documents and folders stored in the QDOC library, if that library is chosen by the user. This does not mean that a user with *ALLOBJ authority will be able to display the document contents. The system will not accept system-assigned object names on commands unless it is shown as an option on the display.

Recommendations for Accessing DLOs

- Do not enroll users in the System Distribution Directory unless specifically needed.
- Exclude users from document library object commands.

13.2.5 Authority

Letting a user with *SECADM special authority create authorization lists means that the user has access to all folders and documents on the system. The owner of the authorization list used to secure documents and folders will have *ALL access to the folders and documents secured by that list unless specifically excluded.

See the section 13.2.4, “Access to Document Library Objects” on page 13-5 for a discussion of *SECADM.

Recommendations for Authority

- Only one or two users should be given authority of *SECADM to ensure smooth running of the application, and both users should be made aware of the responsibilities of that authority.
- Message queues should be secured by public authority *EXCLUDE as messages sent to the user might contain sensitive information.

13.2.6 Access Codes

Access codes can be used to group documents together and to control access to documents for specific users. Access codes are included on the AS/400 for compatibility with earlier systems. (DISOSS and S/38 only).

In order to use access codes, the codes must be created in the system, and users must be authorized to the codes needed. All documents must be secured by the access code chosen for the group that the documents belong to.

Users who are assigned access codes have *USE authority to documents and folders that are assigned the same access code. *USE does not allow a user to edit or change a document or folder. Because of its limited usefulness, you should use access codes only if you used access codes on previous systems and intend to continue using access codes for compatibility.

If you choose to use access code on the system, you should design the arrangement of the codes logically to make it easier to maintain the use of access codes. An example of an implementation of access codes is given in Figure 13-2.

	PAYROLL	RESEARCH	PLANNING	PRODUCTION	LEGAL
TOP SECRET	13	23	33	43	53
SECRET	12	22	32	42	52
CONFIDENTIAL	11	21	31	41	51
UNCLASSIFIED	10	20	30	40	50

Figure 13-2. Possible allocation of Access Codes

A user with access codes 13, 12, 11, 10, 20, 30, 40, 50 could view (remember the access codes only provide *USE authority) all documents of the payroll department, and only unclassified documents of the other departments. Access codes apply only to documents and folders, not to other object types in the system.

The disadvantage of the access codes are their lack of flexibility. If the above user was to be given access to a single confidential document in the production department, a special access code would have to be created, or the user would have access to all documents with access code 34, given access to that code.

Creating new access codes for special purposes can produce an unmanageable security environment and should be avoided.

For the same level of security and a more flexible method of enforcing security, authorization lists should be used. See section 13.2.3, “Authorization Lists” on page 13-4 for a discussion on authorization lists.

Recommendations for Access Codes

- Unless documents are migrated from another system, where access codes are already implemented, **access codes should not be chosen for securing documents**. Group profile authority, authorization lists and specific authority to an object will provide a more flexible security structure that will be easier to maintain

13.2.7 Working on Behalf of Another User

If USER1 gives authority for USER2 to work on his behalf, USER2 must define that he is going to do work for USER1 when he selects an option on the Office main menu. The system will automatically swap the authorities of USER2 with the authorities of USER1 as soon as USER2 defines he is going to work on behalf of USER1. When USER2 is finished working on behalf of USER1 the system will swap USER2′s own authorities back.

After selecting the mail option, USER2 can define for which user he will view mail, by entering the user ID and address of the user. If he is permitted to do work for the user ID, he will see the mail for that user, with the exception of all documents marked “PERSONAL,” which will not be accessible by him. If he is not allowed to do work for the user ID, a message will appear, indicating he is not allowed to do work for this user.

If USER2 is going to do word processing on behalf of USER1, he must select option 5 on the Office main menu, “Documents and folders.” On the following display he must choose option 1 to work with documents and type the user ID and address of USER1 on the prompt line.

If he is not allowed to do work for USER1, a message will appear at the bottom line of the display saying that he is not allowed to do work for this user ID.

If he wants to work with his own documents, he just leaves out the prompt for “Work on behalf of” and he will get to see his own documents.

Recommendations for Working on Behalf of Other Users

- Consider carefully who should be allowed to do work for whom
- Mark all sensitive material as personal with no additional authorities.

13.3 Overview of OfficeVision/400 Security

This section discusses general considerations for security in AS/400 Office. These points are valid for both a standalone system and systems operating in a network. Office security is not a separate scheme on an AS/400 and must be planned together with security considerations for the whole system.

Before setting up security for Office, there are a few fundamental concepts that need to be understood:

- Terms and Definitions for OfficeVision
- Changing User Profiles through Office Enrollment Menu
- Enrolling users
- Limiting Office user options
- Saving procedures
- Authorization lists
- Access to objects outside Office from inside Office
- Distribution lists
- Shared folders

13.3.1 Terms and Definitions for Office

The following are important concepts with regard to Office security:

Indirect User
A person who receives electronic mail, but does not sign on to the system.

Enrolled in Office
An enrolled user can use all the functions in the Office menu he or she is authorized to.

System Distribution Directory

- Automatic enrollment when enrolled in Office
- Enrollment necessary to receive objects from other systems

Distributions
Messages, notes, or documents exchanged between systems.

Distribution Lists
A group of User IDs to receive electronic mail and messages.

Shared Folders
Folders that are shared between PCs and AS/400.

User ID/Address
A two-part network name used in the system distribution directory and in OfficeVision/400 to uniquely identify a user. The network name is usually the same as the user profile name but does not need to be.

13.3.2 Changing User Profiles through Office Enrollment Menu

Office users can be enrolled in three ways:

- Through the Office enrollment menu
- Through the Add Office Enrollment (ADDOFCENR) CL Command
- Through automatic enrollment

In order to enroll Office users, the person responsible for enrollment must have security administration authority. This authority enables the Office administrator to change information on already existing user profiles on the system, when enrolling the user in Office and to create new user profiles. The access to change information on user profiles is not dependent on *ALLOBJ authority, but is dependent on the user profile being a member of a group profile. In order to maintain all Office users′ enrollment information, the Office administrator needs authority to all group profiles, but not to the specific user profiles.

Changing the QSECOFR Profile

An Office administrator can change the following parameters on the QSECOFR user profile:

- Accounting code
- Initial program/library
- Initial menu/library
- Printer
- Message queue/library
- Current library

As the user profile QSECOFR does not belong to a group profile, it will always be possible for the Office administrator to change the QSECOFR user profile.

Changing a User Profile

An Office administrator has access to change all user profiles on the system through the enrollment menu, unless the user profile is a member of a group profile. Under the condition that no group profile is used, the Office administrator can change the following parameters on any user profile except QSECOFR:

- Group profile name
- Accounting code
- Maximum storage
- Limit capabilities
- Initial program/library
- Initial menu/library
- Add special authority *SECADM
- Printer
- Message queue/library
- Current library

Changing a User Profile that Is a Member of a Group Profile

A user profile that is a member of a group profile can be enrolled in Office by an Office administrator without authority to the group profile. The user will be enrolled using the default system values for Office. This means that any company standard that has been set up for naming conventions for folders and calendars will be overridden by the enrollment with system defaults. It will then be the user′s responsibility to change the enrollment information to company standards, and to delete all folders and calendars with names that do not fit the company standard.

In order to allow the Office administrator to maintain enrollment information for the user, the Office administrator must be given the following authorities to the group profile:

- Object operational
- Object management
- Data read
- Data add
- Data update
- Data delete

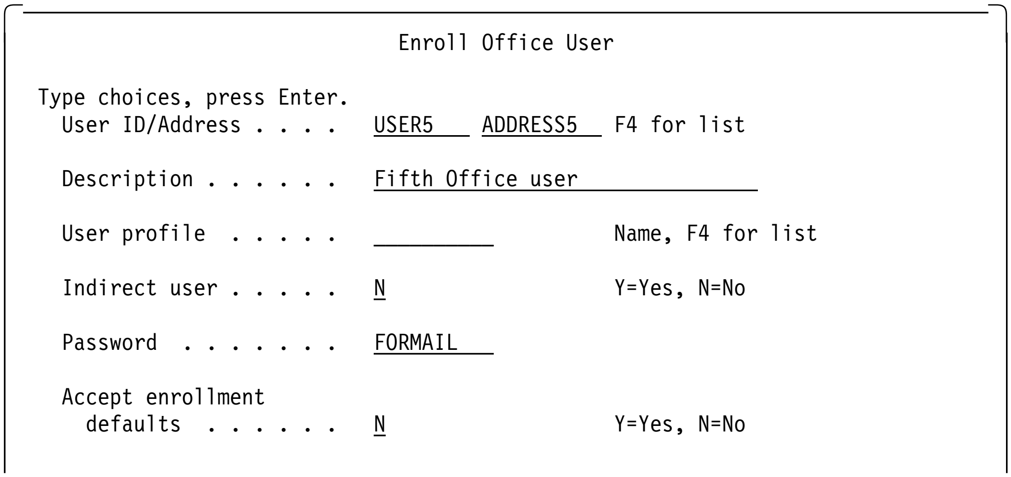
This is equivalent to *CHANGE plus *OBJMGMT authorities. If a user profile belongs to a group profile that the Office administrator does not have access to, the Office administrator will not be able to change information for that user profile.

If the Office administrator is given access to a group profile he can change all the user profiles in the group profile.

Recommendations for Changing User Profiles

- Let all *SECOFR profiles be part of the QSECOFR group profile
- Let *SECOFR profile do the enrollment of Office users
- Do not authorize other Office administrators (if any) to group profiles

13.3.3 Enrolling Users



```
                              Enroll Office User

 Type choices, press Enter.
   User ID/Address . . . .   USER5___  ADDRESS5__  F4 for list

   Description . . . . . .   Fifth Office user________________

   User profile  . . . . .   __________            Name, F4 for list

   Indirect user . . . . .   N                     Y=Yes, N=No

   Password  . . . . . . .   FORMAIL___

   Accept enrollment
     defaults  . . . . . .   N                     Y=Yes, N=No
```

Figure 13-3. Enroll Office User Display

When enrolling a user, the user ID and address combination must be unique on the system. A User ID name can exist several times with different addresses. *Do not confuse the User ID with the user profile.* Although they can be the same, the enrollment process links a given User ID with a user profile. A user profile is used to define a user and the user′s authorities to the local system. A user ID together with an address is used to define either a local user or a user on another system. User ID and address is kept in the System Distribution Directory together with the name of the system that user is known to. The address can be the same as the system name, but does not have to. The address could represent a department, for example. For simplicity and maintainability, we recommend that the user profile and user ID match, and that the address and system name also match.

When a document is sent to a user ID at a certain address, the System Distribution Directory is searched for the combination, and the system name is retrieved to send the distribution.

If enrolling a new user into OfficeVision/400, and creating a user profile at the same time, by default, the new user will have a user profile which matches the user ID. If the user is already a non-Office user on the system, all the appropriate information in the existing user profile will be displayed and can be changed when enrolling the user in Office using the enrollment screens.

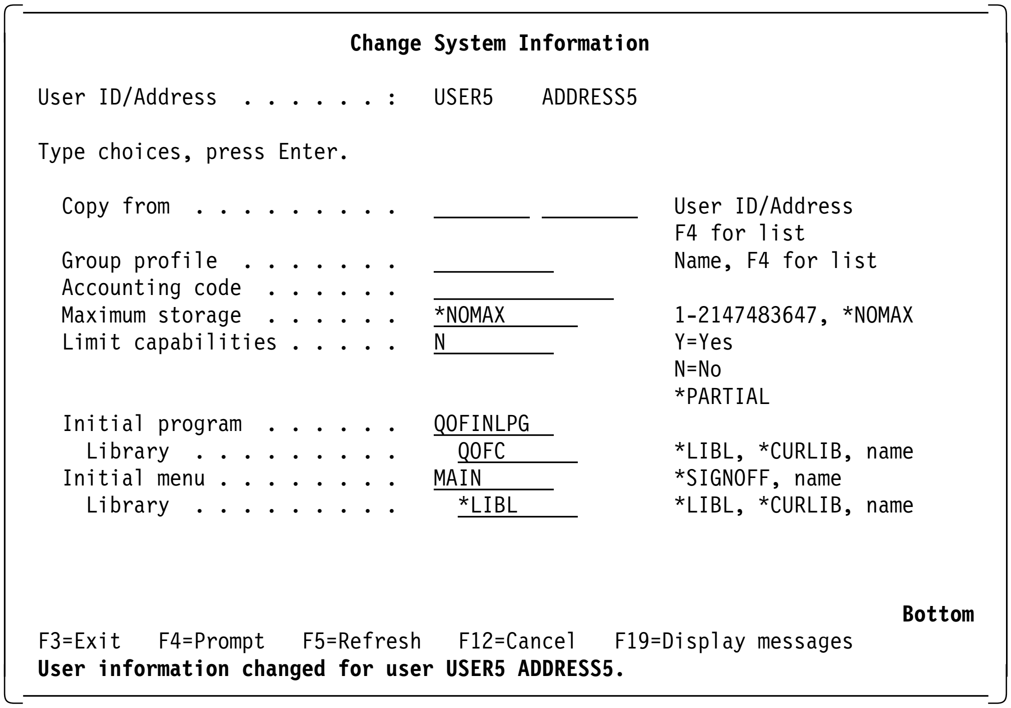
Indirect Users

Indirect users only receive printed mail. A printer to receive the indirect user mail must be defined, and the option to print personal mail will be displayed. Print personal mail should always be *NO. If a non-existent printer name is defined, the mail to be printed will be routed to the system printer.

When a document is printed for a direct user, the first page of the document is a cover sheet indicating who the document is for, who the sender is, what the subject matter is, and any messages that the sender included. If the document is personal it is specified on the cover sheet. If a personal document is sent to an indirect user who has elected not to receive personal mail, only a cover sheet is printed for the user, indicating that an attempt was made to send a personal document to that indirect user. The distribution itself is rejected and an error is sent back to the sender, indicating that the indirect user does not receive personal mail.

If an indirect user elects to receive personal mail, consider the location and security of the printer that will print that mail. If it is in an unsecured area, the security of the document could be compromised.

Enrollment Considerations



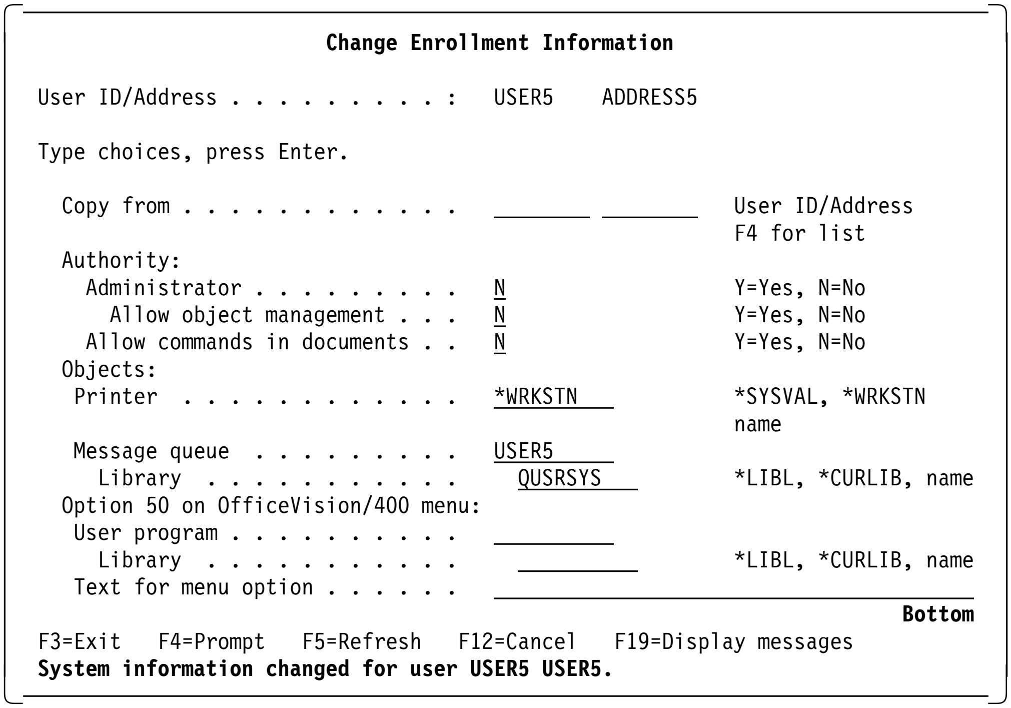
```
                         Change System Information

 User ID/Address  . . . . . . :   USER5    ADDRESS5

 Type choices, press Enter.

   Copy from  . . . . . . . . .   ________ ________   User ID/Address
                                                      F4 for list
   Group profile  . . . . . . .   __________          Name, F4 for list
   Accounting code  . . . . . .   _______________
   Maximum storage  . . . . . .   *NOMAX_____         1-2147483647, *NOMAX
   Limit capabilities . . . . .   N_________          Y=Yes
                                                      N=No
                                                      *PARTIAL
   Initial program  . . . . . .   QOFINLPG__
     Library  . . . . . . . . .     QOFC______        *LIBL, *CURLIB, name
   Initial menu . . . . . . . .   MAIN______          *SIGNOFF, name
     Library  . . . . . . . . .     *LIBL_____        *LIBL, *CURLIB, name



                                                                    Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F19=Display messages
 User information changed for user USER5 ADDRESS5.
```

Figure 13-4. Change System Information Menu

All user profiles on the system should be created with LMTCPB(*YES) unless there is a specific need to be able to use the command line.

If a user needs to have an initial menu or initial program outside Office, the initial menu can be set to include an option to run the command STROFC, or let the program call QOFC/QOFINLPG to direct the user into the Office Main Menu.

For a new user on the system, the Office initial program will be inserted automatically for the prompt “Initial program.” If an existing user is being enrolled, the Office administrator may have to fill in the user′s initial program.



```
                     Change Enrollment Information

User ID/Address . . . . . . . . . :   USER5    ADDRESS5

Type choices, press Enter.

  Copy from . . . . . . . . . . .   ________ ________   User ID/Address
                                                       F4 for list
  Authority:
    Administrator . . . . . . . .   N                  Y=Yes, N=No
      Allow object management . . . N                  Y=Yes, N=No
    Allow commands in documents . . N                  Y=Yes, N=No
  Objects:
   Printer  . . . . . . . . . . .   *WRKSTN___         *SYSVAL, *WRKSTN
                                                       name
   Message queue  . . . . . . . .   USER5_____
     Library  . . . . . . . . . .     QUSRSYS___       *LIBL, *CURLIB, name
  Option 50 on OfficeVision/400 menu:
   User program . . . . . . . . .   __________
     Library  . . . . . . . . . .     __________       *LIBL, *CURLIB, name
   Text for menu option . . . . .   ________________________________________
                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F19=Display messages
System information changed for user USER5 USER5.
```

Figure 13-5. Change Enrollment Information Menu

If enrolling a *SECADM level user into OfficeVision/400, the Administrator prompt will be set to Y by default. A system security administrator is equivalent to an office administrator from a user profile capability viewpoint. When enrolling a new user in Office, if you give that user Administrator capability, this will create the user profile with class *SECADM.

This level of function should be granted carefully, as it gives the user the ability to change profiles of other users. If you wish to allow a user to perform office administration functions, but not to have the capability to change user profiles, then we recommend that the office administrator has a menu to provide different functions, where the menu calls programs which can adopt authority needed to execute the Office CL commands such as:

- Add Office Enrollment (ADDOFCENR)
- Change Office Enrollment (CHGOFCENR)
- Remove Office Enrollment (RMVOFCENR)
- Change Calendar Authority (CHGCALAUT)
- Create Calendar (CRTCAL)
- Delete Calendar (DLTCAL)
- Display Calendar Authority (DSPCALAUT)
- Display Calendar Description (DSPCALD)

Unless the user has a specific need, the prompt for “Allow commands in documents” should be set to “N.” *Allowing commands in a document will enable the user to execute commands from a document that he or she will not be able to otherwise execute*.

If the user works primarily with OfficeVision/400, but uses other programs occasionally, the OfficeVision/400 Main Menu could still be the initial menu for that user. If you specify a user program name and library as in Figure 13-5 on page 13-14 for “Option 50 on OfficeVision/400 menu,” this cause option 50 to appear on the menu, with the associated text.

For users using commands in documents, see 13.3.6, “Access to Objects Outside Office from Inside Office” on page 13-21.

Recommendations for Enrolling Users

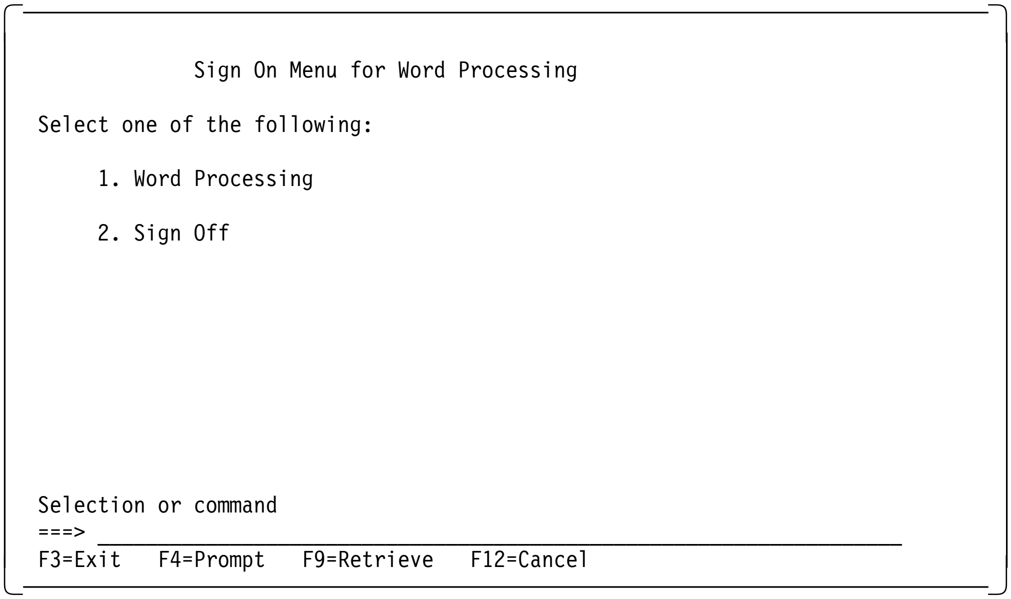
- Indirect users should always be denied the possibility to have personal mail printed.
- Executing commands from a document is like the command “Submit Job” and access to executing commands from a document should be limited to only those who need it.
- If users will access both Office and other system functions, either option 50 should be defined, or an initial menu should be set up for them directing the user into those applications, with OfficeVision/400 being one choice on this menu.
- For Office only users, the initial menu should direct the user into the OfficeVision/400 menu.

13.3.4 Limiting Office User Options

The OfficeVision/400 application can be tailored to meet an installation′s requirements.

Creating a Word Processing-Only Environment

If you want users only to be able to create, revise, print and send documents, you can create your own menu with a single option for Word Processing, and use the WRKDOC command behind this menu option to allow users to access the folders and documents they are authorized to.

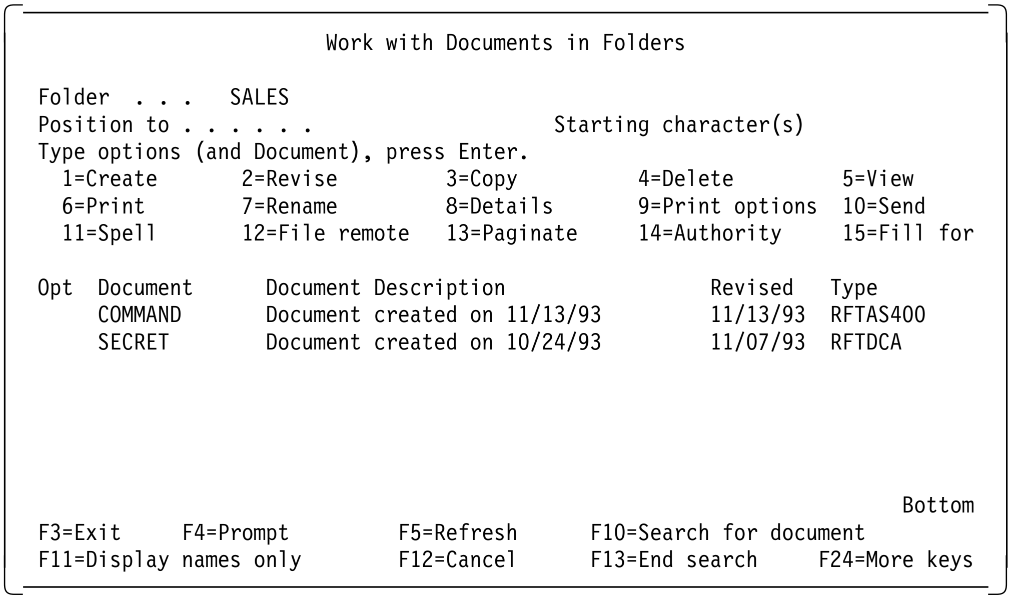


```
              Sign On Menu for Word Processing

 Select one of the following:

      1. Word Processing

      2. Sign Off








 Selection or command
 ===> ____________________________________________________________________
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

*Figure 13-6. User-Created Menu with the Word Processing Option Only. Behind the option is the command WRKDOC with no parameters. The authorization lists in the Office application will decide which folders the user will be able to access and which documents the user will be able to access in the folders. If the command is issued with the parameter *FLR(name), the user will be guided into that folder, otherwise the last used folder will be displayed.*

When selecting option 1, Word Processing, users will be guided directly to the display: "Work with Documents" where they will be able to Create, Revise, Copy, Delete, View, Print, Rename, Change Details, Print with options, Send, Spell check, File remotely, Paginate, Change Authority, and Fill in a form for a document.

This menu could be defined as the initial menu for a user who is only going to perform word processing. If the user is working with other applications, the user should be excluded from all other Office commands in order to ensure that the user will not gain access to other parts of OfficeVision/400. The user profile should specify the LMTCPB(*YES) parameter.



```
                           Work with Documents in Folders

 Folder  . . .   SALES
 Position to . . . . . .                  Starting character(s)
 Type options (and Document), press Enter.
   1=Create       2=Revise        3=Copy          4=Delete        5=View
   6=Print        7=Rename        8=Details       9=Print options  10=Send
   11=Spell       12=File remote  13=Paginate     14=Authority     15=Fill for

 Opt  Document      Document Description                 Revised   Type
      COMMAND       Document created on 11/13/93         11/13/93  RFTAS400
      SECRET        Document created on 10/24/93         11/07/93  RFTDCA






                                                                     Bottom
 F3=Exit     F4=Prompt         F5=Refresh      F10=Search for document
 F11=Display names only        F12=Cancel      F13=End search     F24=More keys
```

Figure 13-7. Work with Documents in Folders Screen. If the user is enrolled in OfficeVision/400 he will have full access to all Office document functions. There will be no difference in performing word processing from an application menu or from the Office main menu.

The user must be enrolled in OfficeVision/400 to be able to create and revise documents. The user will only be able to copy documents if the user only is enrolled in System Distribution Directory, but not in Office.

Combining OfficeVision/400 with Application Programs

Let the user have the Office program as initial program and activate the application program under option 50 on the Office main menu on the user's enrollment record.

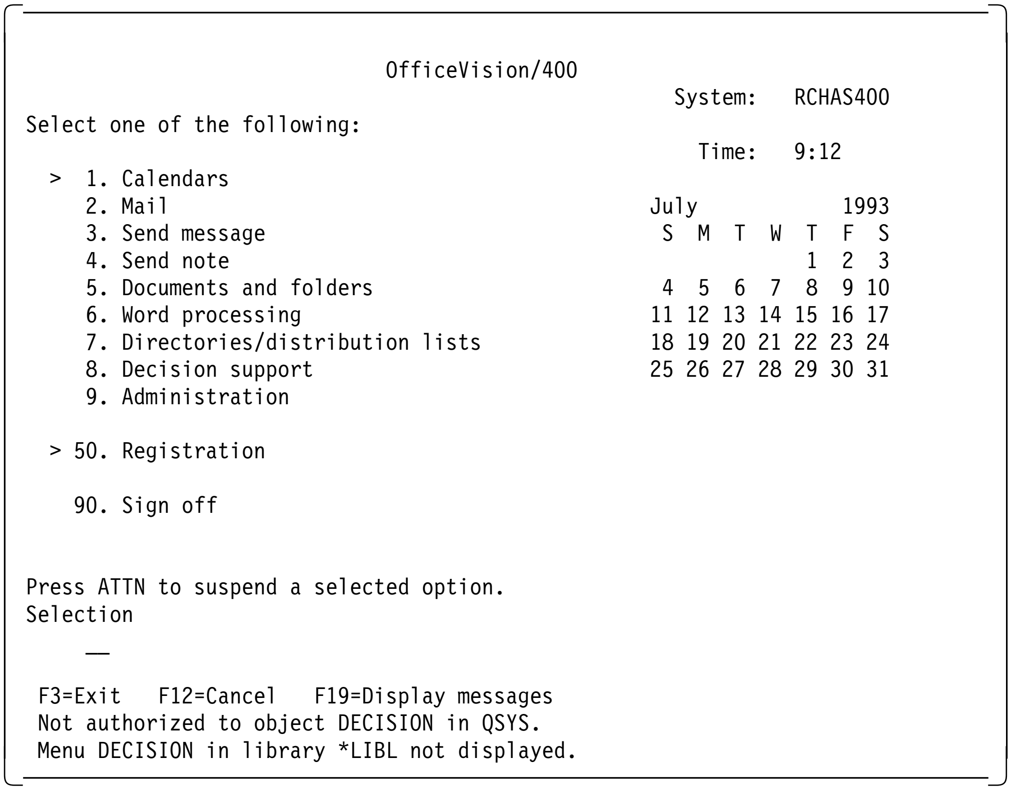
Figure 13-8 on page 13-18 shows the Office main menu with option 50 with the user selected option text.

The two ">" characters beside option 1 and option 50 indicates that the user has selected these options and has suspended them by pressing the Attention key. They are ready for reactivation when chosen.

Exposures of the Decision Support Menu

Option 8 from the OfficeVision/400 main menu allows users to access certain decision support tools. If your objective is that users only have access to Office functions, then their ability to access functions from this menu could be a breach of your desired security structure. This can occur despite the fact that they have LMTCPB(*YES). The Decision Support Menu will allow the user access to IDDU, BGU, Query Utilities, DFU and many other file related commands and menus, just by selecting items on the OS/400 menus. To reduce this exposure, you should consider changing the *PUBLIC authority to the decision menus, or commands and other menus accessed from this menu. This can be done with the GRT/RVKOBJAUT or EDTOBJAUT commands. If the user's authority to Decision menu is *EXCLUDE and option 8 is selected, the following messages are displayed:

```
Not authorized to object DECISION in QSYS.
Menu DECISION in library *LIBL not displayed.
```



```
                                OfficeVision/400
                                                     System:   RCHAS400
Select one of the following:
                                                       Time:   9:12
  >  1. Calendars
     2. Mail                                         July            1993
     3. Send message                                  S  M  T  W  T  F  S
     4. Send note                                                 1  2  3
     5. Documents and folders                         4  5  6  7  8  9 10
     6. Word processing                              11 12 13 14 15 16 17
     7. Directories/distribution lists               18 19 20 21 22 23 24
     8. Decision support                             25 26 27 28 29 30 31
     9. Administration

  > 50. Registration

    90. Sign off


Press ATTN to suspend a selected option.
Selection
     __

 F3=Exit   F12=Cancel   F19=Display messages
 Not authorized to object DECISION in QSYS.
 Menu DECISION in library *LIBL not displayed.
```

Figure 13-8. Message Returned to Office User. When the user tries to choose option 8 "Decision support" without being authorized to the menu DECISION that lies behind that option, the system will return an error message.

13.3.5 Procedures for Saving Office Objects

There are several possibilities for saving Office objects to tape. The SAVDLO command is shown in Table 13-2 on page 13-19. The table shows results and consequences of the parameter selections. Many other variations of parameters can be selected and may prove a combination that will suit your company's needs better than any of these combinations. The keyword *REFCHGDATE refers to the latest date of save, and means that changed that have taken place after that date are saved. The keyword *CHKFORMRK refers to the option on the document description where the user defines if a document is going to be stored off-line in connection with this command or if the document should be left on the system and not be touched by this command.

Parameters	Actions	Consequences	Remarks

DLO(*ALL)
FLR(*ANY)
	All Document Library Objects are saved	Heavy load on system
		Long time to complete
			Should be avoided as daily procedure

DLO(*SEARCH)
FLR(*ANY)
REFCHGDATE(date of save)
	Only changed Document Library Objects are saved	Less load on system
		Less time to complete
		Restoring will need several steps, restoring all changes since the last full save
			Should be combined with weekly full save. This is the recommended approach.

DLO(*SEARCH)
FLR(*ANY)
CHKFORMRK(*YES)
	Only marked Document Library Objects are saved	Documents can be stored off-line minimizing the need for disk space
		If documents are stored off-line they are not easily accessible
		Document descriptions can be kept on the system for document searches
			This should not be the company saving procedure in an environment with many document revisions

Table 13-2. Parameters for the Command SAVDLO

The following objects are associated with mail:

- Distribution Recipient Queue
- Distribution Tracking object
- Distribution Document

The *Distribution Recipient Queue* is an internal object that contains entries for incoming object distributions, incoming document distributions, outgoing document distributions, and error distributions.

The *Distribution Tracking Object* is an internal object that is used to control Office distributions.

These two objects are saved by the SAVSYS command or by the SAVSECDTA command with the parameter MAIL(*YES). Users do not have access to these internal objects.

The **Distribution Document** is an internal document that contains the document content and the document details for distributions. Users do not have access to this internal document. This document is saved by the SAVDLO command with the parameter DLO(*MAIL).

Only a SAVDLO DLO(*ALL) FLR(*ANY) command will save certain "invisible" Office objects to tape. These objects are:

- Distribution lists
- Search results documents

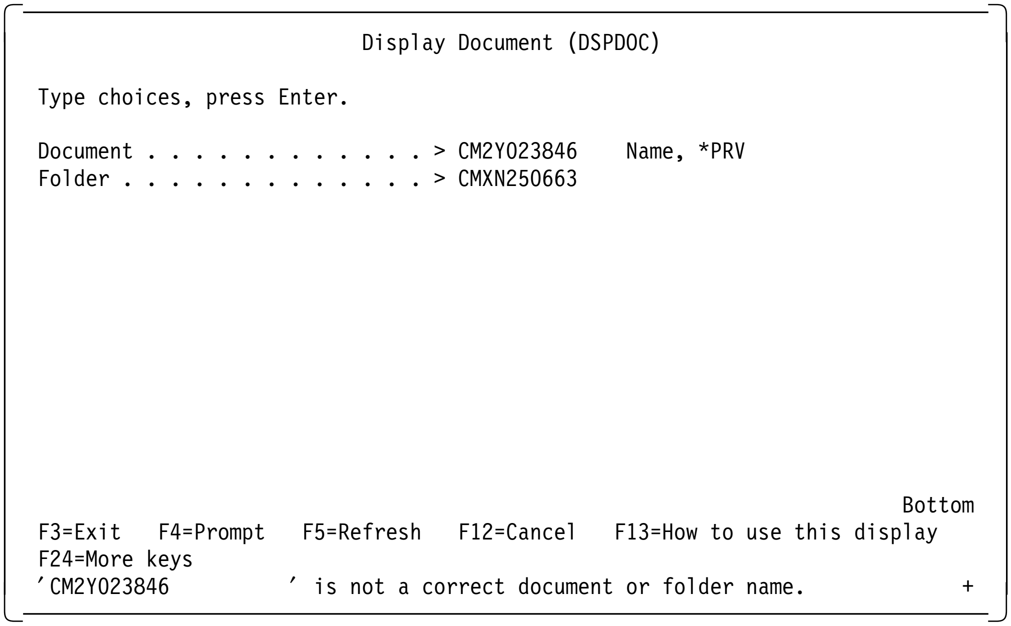
As most of the internal objects associated with mail, distribution lists, and document search lists are not displayed, they are easy to forget when setting up save procedures for the company. As all the objects are vital for the system once set up and running, special care should be taken to ensure complete save of these objects.

Recommendations for Procedures to Save Office Objects

- Company save procedures should be reviewed to ensure total save of all objects associated with Office,
- SAVDLO DLO(*ALL) FLR(*ANY) should be run at least once every week.

How to Find or Enter Names

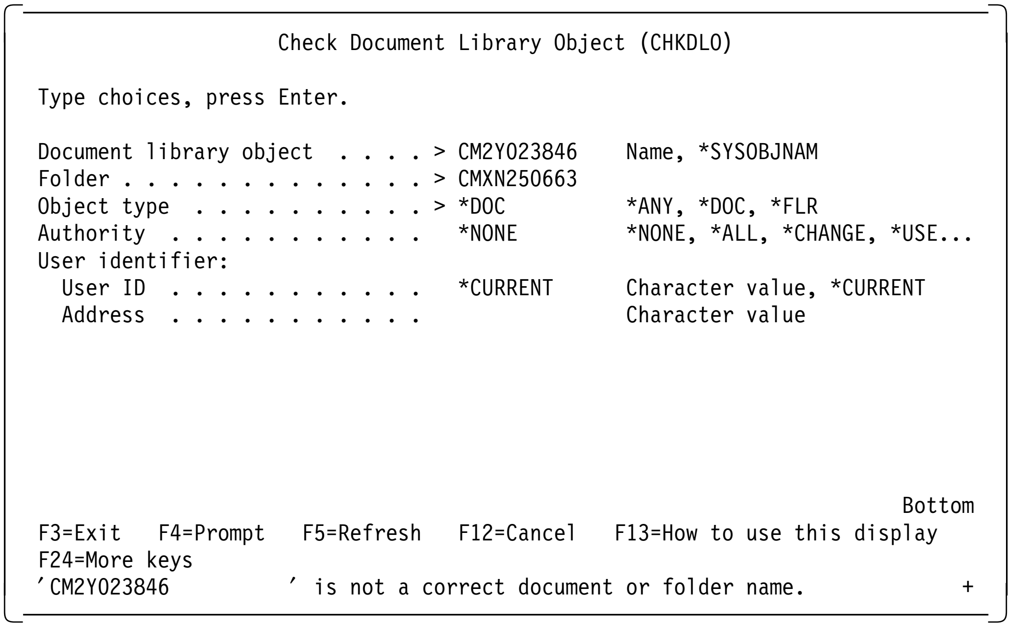
A document has two names. One given by the user who created it, and one assigned by the system. You can use either to find a document.



```
                          Display Document (DSPDOC)

 Type choices, press Enter.

 Document . . . . . . . . . . . . > CM2Y023846    Name, *PRV
 Folder . . . . . . . . . . . . . > CMXN250663

                                                                       Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 'CM2Y023846         ' is not a correct document or folder name.           +
```

Figure 13-9. Display Document Menu

In Figure 13-9, a user trying to display a document by system name is not allowed to do so. If the user had known the correct names of the document and the folder, he or she would have seen the document contents if he or she was authorized to the document.



```
                     Check Document Library Object (CHKDLO)

 Type choices, press Enter.

 Document library object  . . . . > CM2Y023846    Name, *SYSOBJNAM
 Folder . . . . . . . . . . . . . > CMXN250663
 Object type  . . . . . . . . . . > *DOC          *ANY, *DOC, *FLR
 Authority  . . . . . . . . . . .   *NONE         *NONE, *ALL, *CHANGE, *USE...
 User identifier:
   User ID  . . . . . . . . . . .   *CURRENT      Character value, *CURRENT
   Address  . . . . . . . . . . .                 Character value







                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 'CM2Y023846        ' is not a correct document or folder name.            +
```

Figure 13-10. Check Document Library Object Menu

In Figure 13-10 a user trying to use the command CHKDLO with the system name of the folder was not allowed to do so, even though the system accepted the system name for the document.

13.3.6 Access to Objects Outside Office from Inside Office

Part of the word processing of OfficeVision/400 allows access to data on the system through AS/400 QUERY. From a document, it is possible to get to the WORK WITH QUERY display and to create and run queries, besides imbedding them within the document. All the functions of QUERY are available to users accessing QUERY from a document.

It is possible to get to the QUERY main menu, the IDDU main menu, and through these menus to the DFU main menu, by selecting the Decision Support Option on the Office main menu. This means that a user can display or change all information not secured from the user through normal AS/400 resource security.

Limiting users from these options can be done either at the command level or by excluding the users from the menu object DECISION. There is no way to exclude the option DECISION from the Office menu.

If a user has a need to access data through a QUERY, the level of authority to the files and libraries necessary for the user′s work should be identified and granted for each specific object.

Users can, if permitted to do so in the Office enrollment record, include commands in documents. If a user is allowed to include commands in a document, specific authority for the user and the command should be established. All commands available to a user with LMTCPB(*NO) are available to users when allowed to include commands in documents. The user′s need for authority to commands should be carefully reviewed and the authority should be given for the specific command.

If a user who is not allowed to use commands in documents tries to use a command, the document will not print and there will be an entry in QHST and QAUDJRN (if active) that an invalid attempt was made to use a command.

If a user who is allowed to use commands in documents tries to use a command that he is not authorized to, the document will not print and there will be an entry in QHST and QAUDJRN (if active) that an attempt was made to use a command that the user was not authorized to.

Recommendations for Accessing External Office Objects

- Grant a user authority to data specific for the data and for the user:
  - Using one group profile to grant authority to all Office users, will give all Office users included in the profile the same authority to the same data.
  - If the Office user is included on an application-group-profile, and that profile′s authority is used for accessing data through QUERY, the user might have better access to data through the QUERY than he would have, being restricted through menus and programs like non-Office users.
- If a user has a need to imbed CL-commands in a document, authority should be granted to the user for the specific command. Using a command in a document will give the user full access to all parameters in that command and will not be limited through the lack of the command line.

13.3.7 Distribution Lists

When creating entries in the System Distribution Directory, the entry should show whether a user is enrolled in Office or not and whether the user is a direct or indirect user. This is the only way to inform users creating distribution lists that entries in the System Distribution Directory should be treated with special consideration.

For example, non-Office users, included in a distribution list used for personal mail will never get to see their messages. There will be an entry on their message queue, saying a personal distribution has arrived, but they do not get to see the contents of the distribution.

An indirect user on a distribution list used for distributing sensitive material will have his copy printed on a printer for everyone to see.

Recommendations for Distribution Lists

- Mark indirect users clearly on distribution lists
- Use distribution list names that are meaningful

13.3.8 Shared Folders

The shared folders function is used in PC Support. Even if no PC user is using Office, the Office security should be applied to these folders to prevent accidental loss of data.

To the AS/400, a PC Support folder is treated no differently than a normal document folder and it will be included in the SAVDLO commands if security for the folder is established correctly.

If PC users are using OfficeVision/400, they are not distinguished from local users and no special consideration should be taken with regards to Office. Special consideration should be taken in other areas, however. These considerations are covered in Chapter 12, “PC Support” on page 12-1.

13.4 AS/400 Local Functions

In this section, the functions of OfficeVision/400 will be discussed. All functions take place within one AS/400. 13.5, “AS/400 Exchanging Distributions with Remote Systems” on page 13-34 will discuss functions related to interconnected systems.

The following features are considered:

- Create folders
- Create/revise documents/notes/messages
- Send/receive messages
- Send/receive notes
- Send/receive documents
- Managing calendars

13.4.1 Create Folders

A user (without special authorities) may create new folders. New folders will be owned by the user who created them, and public authority is set to *EXCLUDE. Users can be limited in creating new folders. See 13.3.3, “Enrolling Users” on page 13-12 and 13.2.1, “Creating Folders” on page 13-3.

Folders can be kept in other folders, in a structure similar to a PC subdirectory.

For example, a folder COMMON contains two folders PUBLIC and CONFID. Folder PUBLIC contains publicly accessible documents and folder CONFID contains confidential documents.

Folder CONFID contains two other folders, DEPT and MAN, containing departmental documents and management documents. Folder DEPT further contains folder DEPTA, containing monthly bulletins for department ′A′.

With the correct authorization, a user can reach document DEC93 by selecting folder COMMON, then folder CONFID, folder DEPT, folder DEPTA and finally the document DEC93. Another way to reach the document could be to type the complete folder path, when using the option to Work with documents in folders, as follows:

```
COMMON/CONFID/DEPT/DEPTA
```

The list of documents in the folder DEPTA will be displayed. *The user does not need to be authorized to all the folders in the folder path, to access a document in a folder to which he is authorized*, as explained in Figure 13-11 on page 13-24.

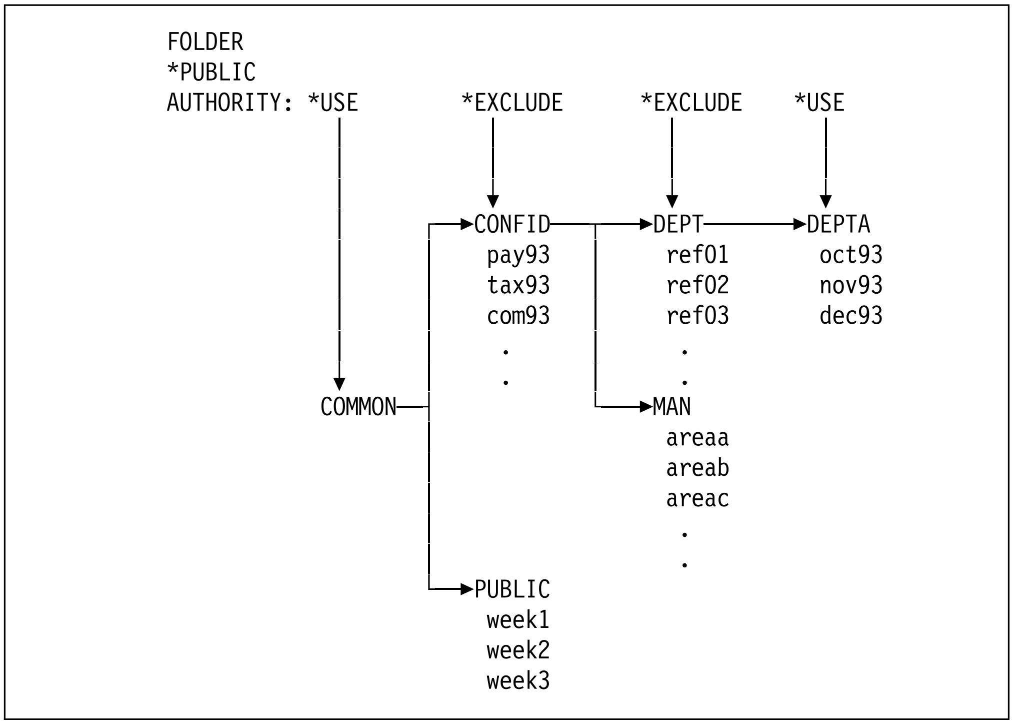
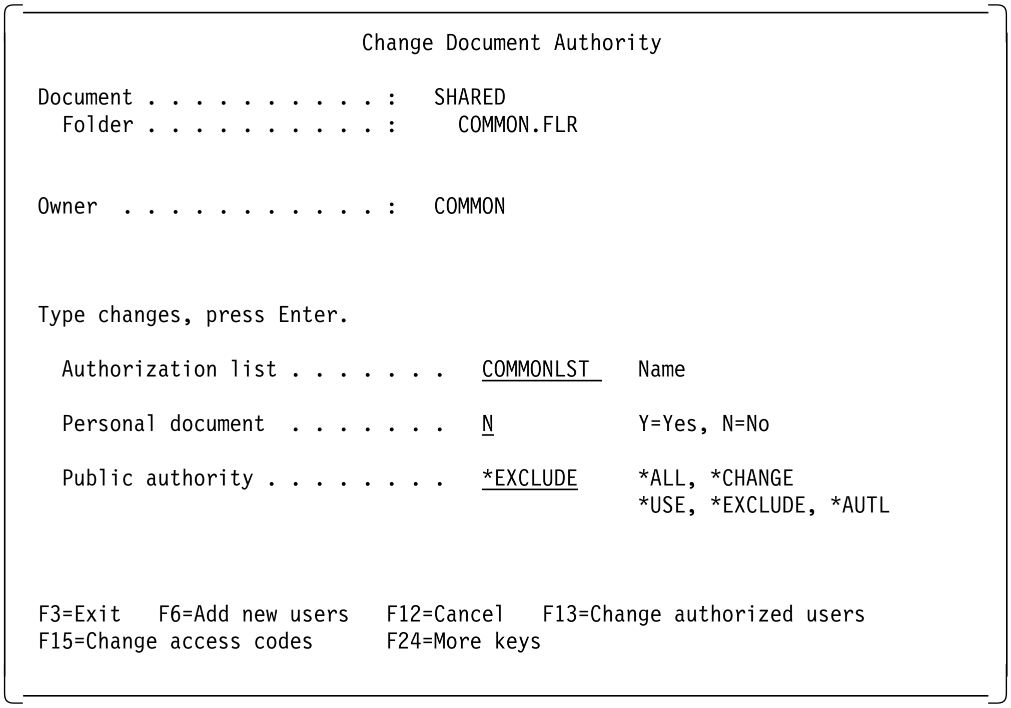


*Figure 13-11. Folder Authority. A user needs to access document DEC93 in folder DEPTA. The user has no special authorities. When 'work with documents in folders' is used, the folder path is COMMON/CONFID/DEPT/DEPTA. Folders COMMON and DEPTA have *PUBLIC *USE authority. Folders CONFID and MAN are *PUBLIC *EXCLUDE authority. If the user enters COMMON/CONFID/DEPT the message 'request not allowed with folder', will be returned. Note that if the user tries to access the document from the 'work with folders' option, he would need *PUBLIC *USE authority to the previous folder in the path (folder DEPT), in order to access folder DEPTA.*

In Figure 13-12 on page 13-25 a convention of the user's name followed by a description of the object is shown. The folder name is "COMMON" and "FLR" and the authorization list name is "COMMON" and "LST."

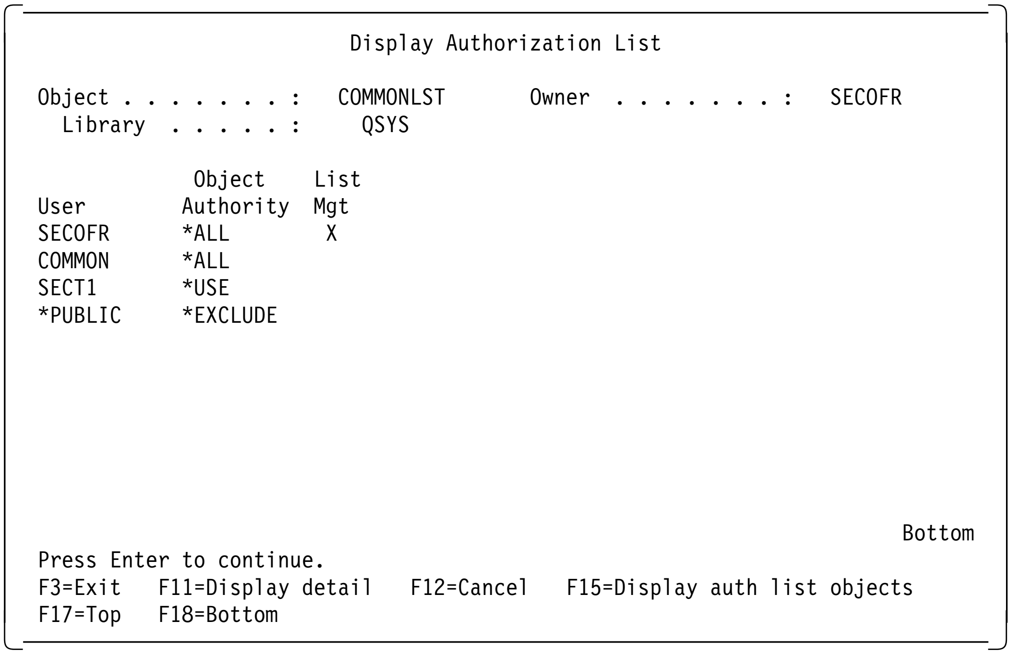


```
                             Change Document Authority

Document . . . . . . . . . . :   SHARED
  Folder . . . . . . . . . . :     COMMON.FLR


Owner  . . . . . . . . . . . :   COMMON



Type changes, press Enter.

  Authorization list . . . . . . .   COMMONLST     Name

  Personal document  . . . . . . .   N             Y=Yes, N=No

  Public authority . . . . . . . .   *EXCLUDE      *ALL, *CHANGE
                                                   *USE, *EXCLUDE, *AUTL



F3=Exit   F6=Add new users   F12=Cancel   F13=Change authorized users
F15=Change access codes      F24=More keys
```

Figure 13-12. Change Document Authority Screen

The user creating a folder is responsible for giving *ALL authority to the user responsible for the backup of the folder, unless that the user has *SAVSYS special authority. Not having this authority means that the backup will fail and it will not be possible to recreate the folder contents if needed. Granting the user responsible for saving the folder to tape *SPCAUT(*SAVSYS) would be sufficient.

The user can give access to the folder by an authorization list. The folder in Figure 13-12 is secured by authorization list COMMONLST.



```
                           Display Authorization List

Object . . . . . . . :   COMMONLST       Owner  . . . . . . . :   SECOFR
  Library  . . . . . :     QSYS

            Object    List
User        Authority  Mgt
SECOFR      *ALL        X
COMMON      *ALL
SECT1       *USE
*PUBLIC     *EXCLUDE









                                                                     Bottom
Press Enter to continue.
F3=Exit   F11=Display detail   F12=Cancel   F15=Display auth list objects
F17=Top   F18=Bottom
```

Figure 13-13. Display Authorization List Screen

In order to read or copy documents from the folder, users must have *USE authority to the document. *Planning for and Setting Up OfficeVision/400* has a detailed overview of the levels of authority needed to perform operations on folders and documents.

In order to create new documents in the folder, users must have *CHANGE authority to the folder.

As shown in Figure 13-13 on page 13-25, the users SECOFR and COMMON can save the folder COMMON.FLR, secured by authorization list COMMONLST, and can create new documents in the folder. The user SECT1 can copy or read documents in the folder COMMONFLR, but will not be able to create new documents. No other user is allowed to work with the folder COMMONFLR.

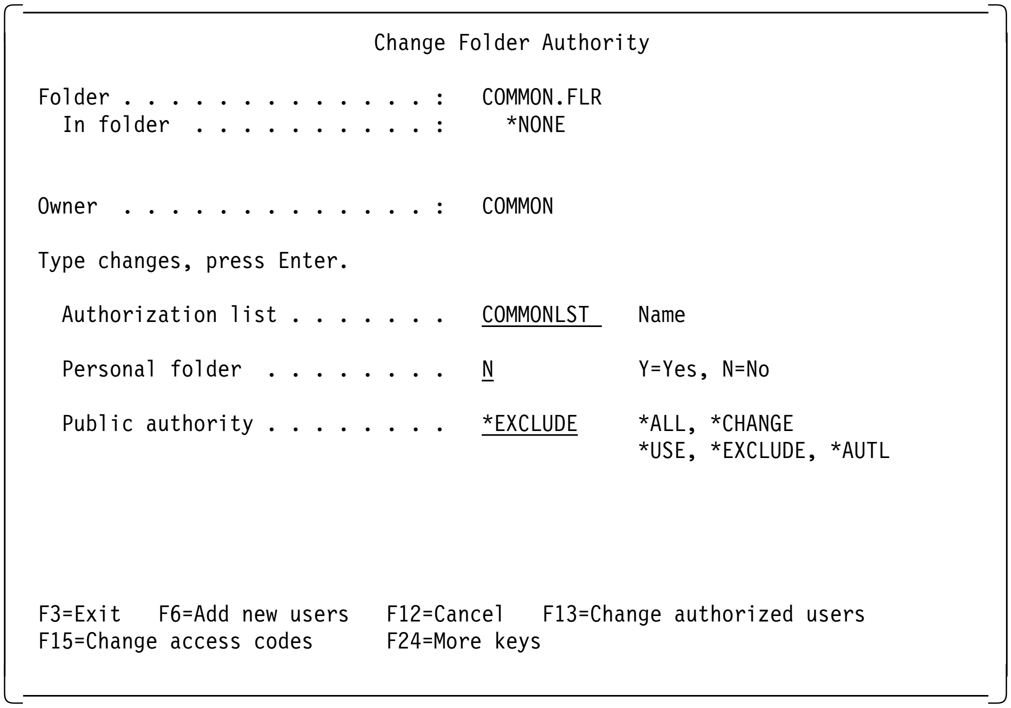
Recommendations for Creating Folders

- Naming conventions should be established to prevent confusion among the users.
- If a fixed folder structure is to be maintained, users at least should be excluded from the option to create new first-level folders. Then all first-level folders must be created by *SECADM or *SECOFR. See 13.2.1, "Creating Folders" on page 13-3 for more recommendations on this.
- All folders should be secured by authorization lists.

13.4.2 Creating and Revising Documents

When a user creates a new document in his own folder, this document will have the same authorities defined as the folder. If a user wants to change authorities to a document, he must create it and then change authorities. If the user later changes the authorities of the folder, the documents already in the folder will keep the authorities initially defined. New documents will have the new authorities.

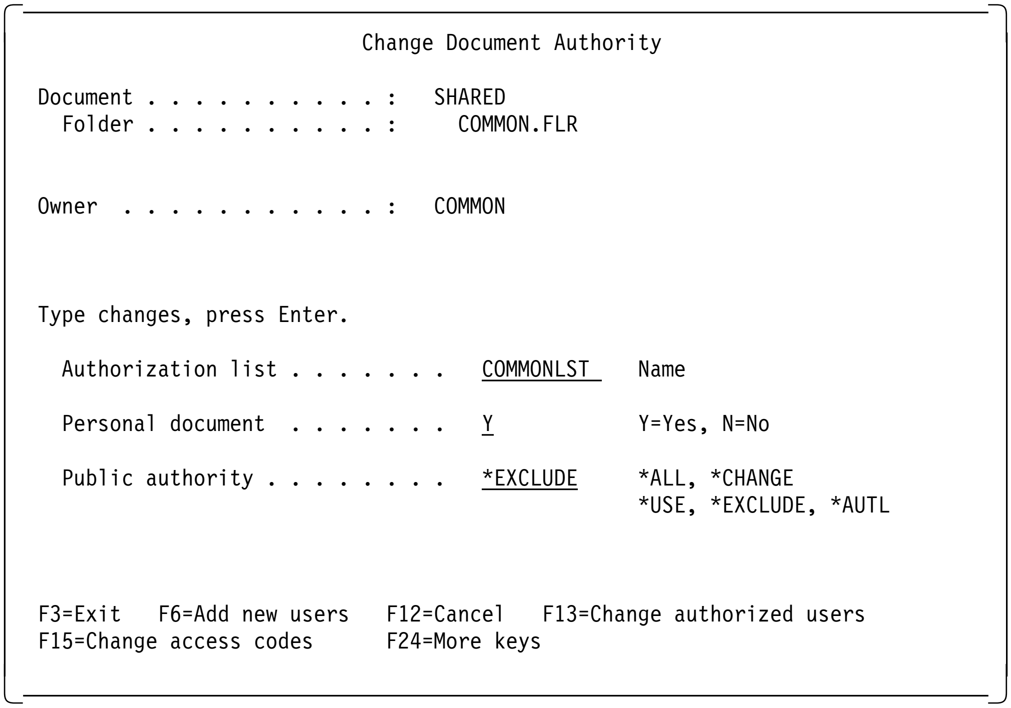
Users must have *USE authority to a document to be able to read it or copy it to their own folders, and have *CHANGE authority to be able to revise it, when it is still in another user′s folder unless they are the owner.



```
                           Change Folder Authority

Folder . . . . . . . . . . . . . :   COMMON.FLR
  In folder  . . . . . . . . . . :     *NONE


Owner  . . . . . . . . . . . . . :   COMMON

Type changes, press Enter.

  Authorization list . . . . . . .   COMMONLST     Name

  Personal folder  . . . . . . . .   N             Y=Yes, N=No

  Public authority . . . . . . . .   *EXCLUDE      *ALL, *CHANGE
                                                   *USE, *EXCLUDE, *AUTL




F3=Exit   F6=Add new users   F12=Cancel   F13=Change authorized users
F15=Change access codes      F24=More keys
```

Figure 13-14. Change Folder Authority Screen

It is possible to combine authorization list authority with specific authority to a document. Figure 13-13 on page 13-25 shows that user SECT1 has *USE authority on the authorization list. In Figure 13-14 *F6=Add new users* can be used to override authorization list security for a single document.



```
                          Change Document Authority

Document . . . . . . . . . . . :   SHARED
  Folder . . . . . . . . . . . :     COMMON.FLR


Owner  . . . . . . . . . . . . :   COMMON



Type changes, press Enter.

  Authorization list . . . . . . .   COMMONLST     Name

  Personal document  . . . . . . .   Y             Y=Yes, N=No

  Public authority . . . . . . . .   *EXCLUDE      *ALL, *CHANGE
                                                   *USE, *EXCLUDE, *AUTL



F3=Exit   F6=Add new users   F12=Cancel   F13=Change authorized users
F15=Change access codes      F24=More keys
```

Figure 13-15. Change Document Authority Screen with Personal Document

In Figure 13-15, the document SHARED is secured by the authorization list COMMONLST. Users permitted to work for user COMMON will adopt user

COMMON′s authorities in the Office part of the system. This adoption of authority can take place in connection with mail processing and Work with documents in folders. Because user COMMON has secured document SHARED by making it personal, other users working on behalf of user COMMON will not have access to this document.

Users included on the authorization list used to secure document SHARED will have access to the document, even if it is marked as personal.

Recommendations for Creating and Revising Documents

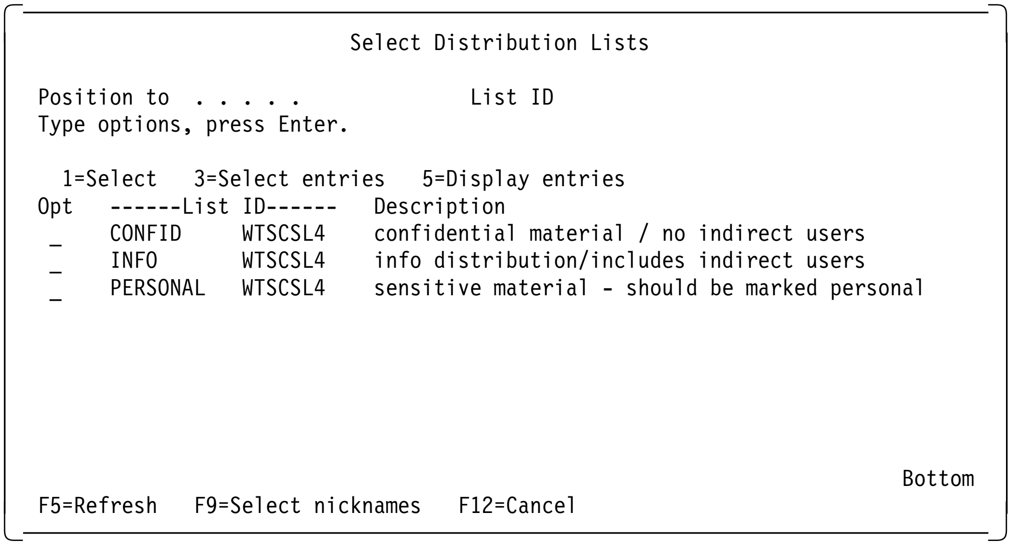
- Use the same authorization list for documents and the related folders.
- Group documents into folders according to subject and authorization.
- Do not automatically give everyone *CHANGE authority to documents in a folder. *USE will enable other users to copy the document into another folder.

13.4.3 Sending Messages, Notes and Documents

Once a user is enrolled in Office he can use the electronic mail function. No further enrollment is needed.

In order to send a message, a note or a document to another user, either on the local system or on a remote system, the user must know the identity of the receiving user. This user ID is stored in the System Distribution Directory. The user ID can either be typed each time it is used, or the user ID can be included on a distribution list.

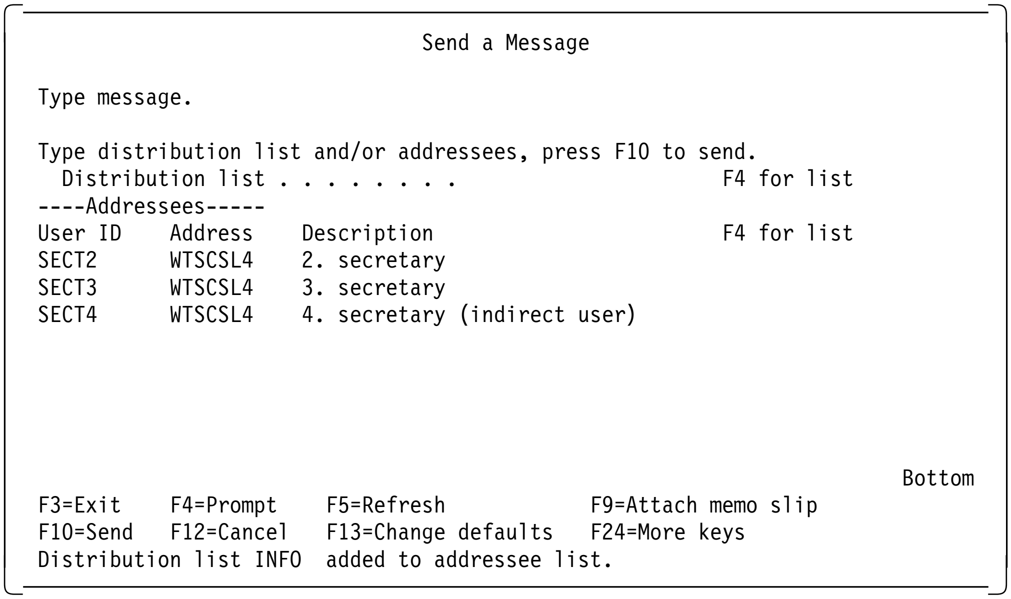
You can create distribution lists to ease distribution to multiple users. Using distribution lists can prevent error messages when a user ID or an address is misspelled.



```
                          Select Distribution Lists

 Position to  . . . . .              List ID
 Type options, press Enter.

   1=Select   3=Select entries   5=Display entries
 Opt   ------List ID------   Description
  _     CONFID     WTSCSL4    confidential material / no indirect users
  _     INFO       WTSCSL4    info distribution/includes indirect users
  _     PERSONAL   WTSCSL4    sensitive material - should be marked personal







                                                                      Bottom
 F5=Refresh   F9=Select nicknames   F12=Cancel
```

Figure 13-16. Select Distribution Lists Menu

When sending a document, a note, or a message, pressing F4 on the distribution list selection prompt will enable the user to select a distribution list. In Figure 13-16, the user can choose between three distribution lists, and all three of them clearly mark information about indirect users. The user responsible for creating distribution lists should use a similar marking method to prevent confusion about the use of the distribution list. Setting up names and descriptions of distribution lists will help the user to decide which list to select.



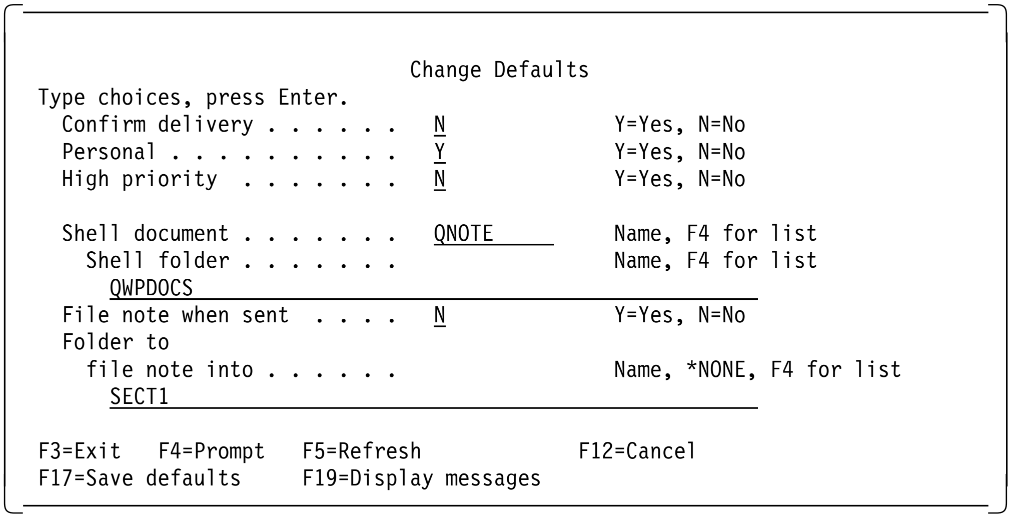
```
                                Send a Message

  Type message.

  Type distribution list and/or addressees, press F10 to send.
    Distribution list . . . . . . . .                     F4 for list
  ----Addressees-----
  User ID    Address    Description                       F4 for list
  SECT2      WTSCSL4    2. secretary
  SECT3      WTSCSL4    3. secretary
  SECT4      WTSCSL4    4. secretary (indirect user)






                                                                         Bottom
  F3=Exit    F4=Prompt    F5=Refresh           F9=Attach memo slip
  F10=Send   F12=Cancel   F13=Change defaults   F24=More keys
  Distribution list INFO  added to addressee list.
```

Figure 13-17. Send a Message Menu

Selecting the distribution list will display the users together with a brief description. Users can be deleted from, or added to, the addressee list before sending the document, the note, or the message.

Users can permit other users to work with their mail. When sending sensitive material, users should classify the mail personal, in order to prohibit unauthorized access to the document.

Pressing F13 on the display shown in Figure 13-17 (change defaults) will enable the user to make the distribution personal.



```

                               Change Defaults
  Type choices, press Enter.
    Confirm delivery . . . . . .  N             Y=Yes, N=No
    Personal . . . . . . . . . .  Y             Y=Yes, N=No
    High priority  . . . . . . .  N             Y=Yes, N=No

    Shell document . . . . . . .  QNOTE         Name, F4 for list
      Shell folder . . . . . . .                Name, F4 for list
        QWPDOCS
    File note when sent  . . . .  N             Y=Yes, N=No
    Folder to
      file note into . . . . . .                Name, *NONE, F4 for list
        SECT1

  F3=Exit   F4=Prompt   F5=Refresh            F12=Cancel
  F17=Save defaults     F19=Display messages
```

Figure 13-18. Change Defaults Menu when Sending a Note

When pressing F13 on the display shown in Figure 13-17 the display shown in Figure 13-18 will appear. On this display the user can request *Confirmation of Delivery* and make the distribution personal.

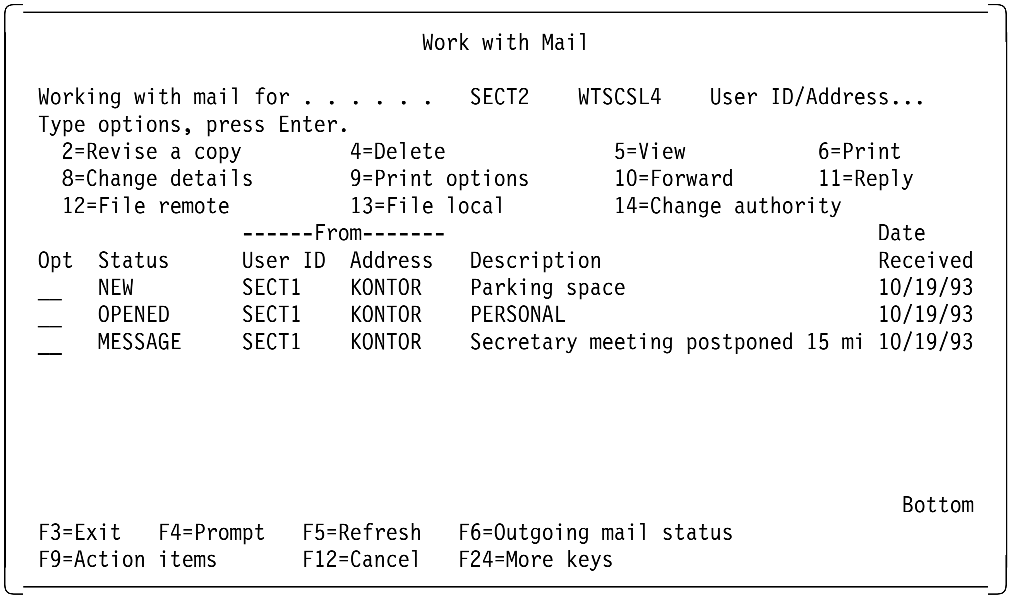
The lower half of the display will vary depending on the sort of distribution the user is going to send. In this case the user is going to send a note and can make the following selections at the bottom of the display:

- Save the note in a folder
- The name of the folder to save the note in
- The shell document for the note, that is, the empty note format to be used for this note

The Office administrator can change the IBM supplied shell note or create a new shell note. If the Office administrator has created a new shell note, the name of that shell note should be typed instead of accepting the IBM delivered shell note. The user-created shell note can be stored in any folder. A user′s note can be stored in any folder.

13.4.4 Receiving Messages, Notes, and Documents

When mail has been sent to a user, there will be a message informing the user of the arrival of new mail on the Office main menu. When the user selects the option Work with mail, the panel shown in Figure 13-19 is displayed. This is the display for incoming mail. Personal mail is clearly shown as Personal on the display. There is a distinction between messages, notes, and documents, as can be seen from the Work with mail display. If the user wants to see the status for outgoing mail, he will have to press F6 on the display. The distribution can be forwarded by selecting option "10" in front of the distribution to be sent. Additional text can be typed in the note and new recipient information.



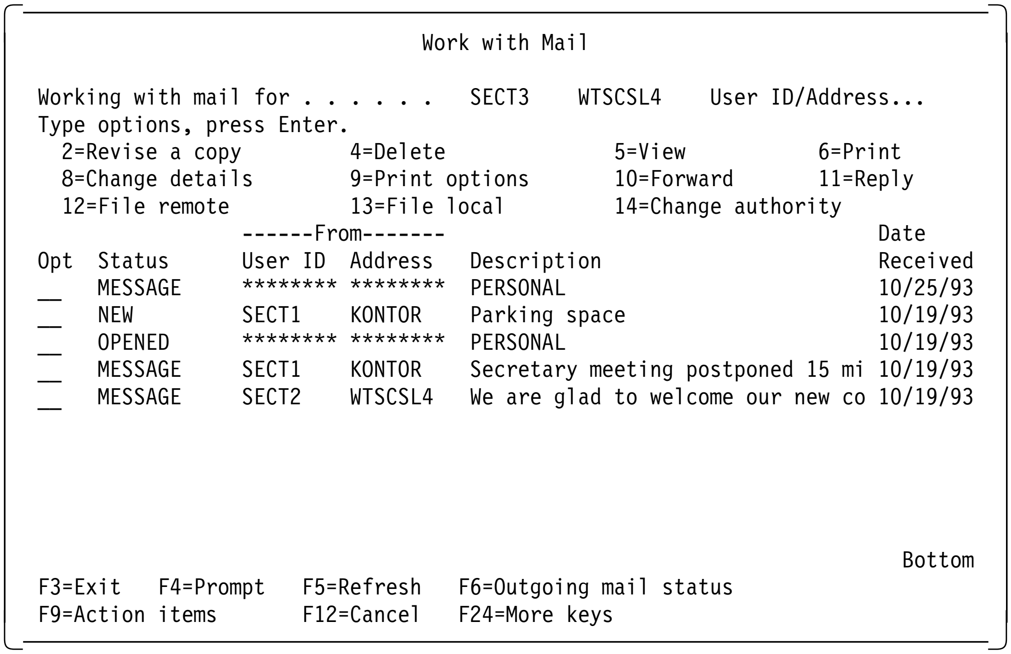
```
                              Work with Mail

Working with mail for . . . . . .   SECT2    WTSCSL4    User ID/Address...
Type options, press Enter.
  2=Revise a copy        4=Delete              5=View          6=Print
  8=Change details       9=Print options       10=Forward      11=Reply
  12=File remote         13=File local         14=Change authority
                 ------From-------                                 Date
Opt  Status      User ID  Address   Description                    Received
__   NEW         SECT1    KONTOR    Parking space                  10/19/93
__   OPENED      SECT1    KONTOR    PERSONAL                       10/19/93
__   MESSAGE     SECT1    KONTOR    Secretary meeting postponed 15 mi 10/19/93





                                                                     Bottom
F3=Exit   F4=Prompt   F5=Refresh   F6=Outgoing mail status
F9=Action items       F12=Cancel   F24=More keys
```

Figure 13-19. Work with Mail Menu, User′s Own Mail Log

In Figure 13-19, a user was working with his own mail. Even in the user′s own mail log, personal mail is clearly marked.

When working on another user′s behalf, the user will swap authorities with the other user as long as he works on his behalf. This is also valid when the Office administrator works on behalf of other users.

When the user is working with mail on another user′s behalf, he can view and forward the other user′s mail as if it was his own, but he cannot work with the other user′s personal mail. He can see that a personal distribution has arrived and what type it is (message, document or note), but he cannot see who sent it or identify the subject for the distribution.



```
                                Work with Mail

Working with mail for . . . . . .   SECT3    WTSCSL4    User ID/Address...
Type options, press Enter.
  2=Revise a copy         4=Delete              5=View          6=Print
  8=Change details        9=Print options       10=Forward      11=Reply
  12=File remote          13=File local         14=Change authority
                 ------From-------                                    Date
Opt  Status      User ID  Address   Description                       Received
__   MESSAGE     ******** ********  PERSONAL                          10/25/93
__   NEW         SECT1    KONTOR    Parking space                     10/19/93
__   OPENED      ******** ********  PERSONAL                          10/19/93
__   MESSAGE     SECT1    KONTOR    Secretary meeting postponed 15 mi 10/19/93
__   MESSAGE     SECT2    WTSCSL4   We are glad to welcome our new co 10/19/93





                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F6=Outgoing mail status
F9=Action items       F12=Cancel   F24=More keys
```

Figure 13-20. Work with Mail Menu, Working on Other User′s Behalf

Recommendations for Receiving Messages, Notes and Documents

- Give distribution lists meaningful names
- Mark distribution lists that contain indirect users
- Mark sensitive distributions personal

13.4.5 Sending a Message

Sending a message to an Office user will cause the message to be displayed on the user′s mail-log and on the message queue. Sending a message to a non-Office user will cause the message to be displayed on the user′s message queue.

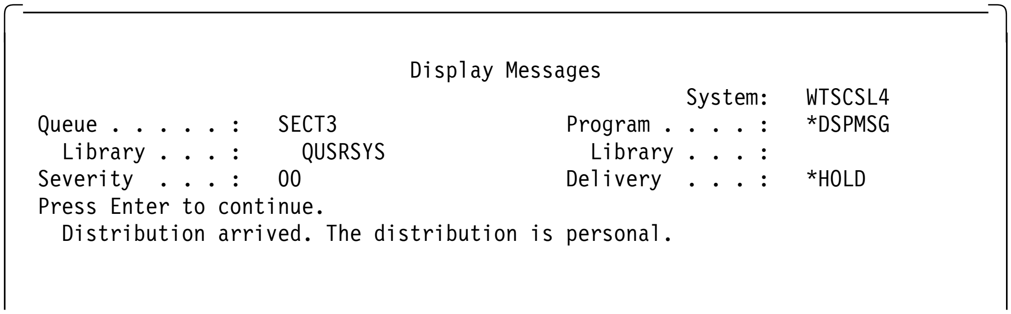
Sending a personal message to an Office user will cause the message to be displayed as personal on the user′s mail log. There will be a message on the message queue saying a personal distribution has arrived. Sending a personal message to a non-Office user will cause a message to be displayed on the user′s message queue, saying that a personal distribution has arrived. The non-Office user cannot get to see the message, because he is not enrolled in Office and has no mail log. When the owner of the queue displays the message queue he will not be allowed to see the body of the note.

Recommendations for Sending Messages

- Do not use personal distribution in connection with messages
- Do not distribute sensitive information in the form of a message

13.4.6 Receiving a Message

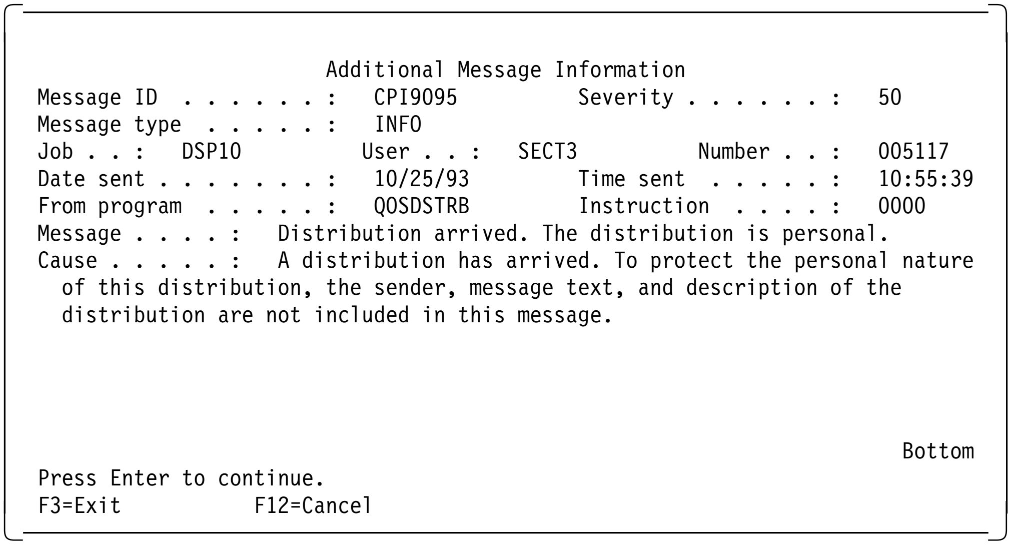
When a message arrives at a user′s mail log, it is displayed as shown in Figure 13-19 on page 13-30. From the mail log the user can view, forward, receive or delete the message. At the same time there will be an entry in the user′s message queue, as shown in Figure 13-21 for the personal message. A message that is not personal will be displayed on the message queue.



```
                             Display Messages
                                                       System:   WTSCSL4
Queue . . . . . :   SECT3                 Program . . . . :   *DSPMSG
  Library . . . :     QUSRSYS               Library . . . :
Severity  . . . :   00                    Delivery  . . . :   *HOLD
Press Enter to continue.
  Distribution arrived. The distribution is personal.
```

Figure 13-21. Display Messages Screen with Personal Message

Pressing HELP on the personal message shown in Figure 13-21 will give the following display:



```
                         Additional Message Information
Message ID  . . . . . . :   CPI9095          Severity . . . . . . :   50
Message type  . . . . . :   INFO
Job . . :   DSP10          User . . :   SECT3          Number . . :   005117
Date sent . . . . . . . :   10/25/93         Time sent  . . . . . :   10:55:39
From program  . . . . . :   QOSDSTRB         Instruction  . . . . :   0000
Message . . . . :   Distribution arrived. The distribution is personal.
Cause . . . . . :   A distribution has arrived. To protect the personal nature
  of this distribution, the sender, message text, and description of the
  distribution are not included in this message.




                                                                     Bottom
Press Enter to continue.
F3=Exit           F12=Cancel
```

Figure 13-22. Additional Message Information Screen with Personal Message

The user will then have to go to the display “Work with Mail” to be able to see the message. The body of the message will be kept in a system internal file. It is not possible for the user to access this file.

Recommendations for Receiving Messages

- Message queues should be secured the same way as other objects to prevent unauthorized access to the queue.

13.4.7 Sending Notes

When sending a note you can also secure it by making it personal.

13.4.8 Receiving Notes

When a user receives a note or a document, there is no clear indication on the mail log whether it is a note or a document. See Figure 13-19 on page 13-30 for the display. All the options on the display are valid for both notes and documents.

For a user working on another user′s behalf the panel shown in Figure 13-20 on page 13-31 will be shown. Information regarding the personal note will not be displayed on the mail log. Users working on behalf of the receiver will not have access to the note.

13.4.9 Sending Documents

You can send documents to other Office users on the system, and to indirect users.

If you send secured documents to other users, these users become owners of their own copy of the document. They can then do with the document what they want to. Documents sent to indirect users will be printed on designated printers.

13.4.10 Receiving Documents

When users receive a document in their mail log, the copy of the document will be theirs. They can store it in a folder and then secure it by making it personal. They can decide to forward a copy to another user, who will then be the owner of his document. There is no way to ensure that a document sent to one user will not be forwarded to another user. When receiving the document the receiver can change authorities to the document. The sender of the document will have *ALL authority to the document, but if he does not have access to the folder in which the document is stored, he cannot get to the document.

13.4.11 Calendars

When a user is enrolled in Office, a calendar is set up for the user. The user is the owner and manager of that calendar. At the same time, the access that other users have to the calendar is determined. A user can create more calendars and revise other user′s authority to his calendars. A user with *SECADM can change the authority to, and description for, any calendar in the system, but will not be able to view or change calendar items unless specifically authorized to do so.

Calendars can only be managed from the WORK WITH CALENDAR display. Calendars can only be saved one by one, and only by the owner, by a person with *SECADM or *SAVSYS special authority.

When a user gives another user access to his calendars, he can allow the other user to view, enter items or change items in his calendar. Allowing another user to enter items in a calendar will also allow the user to change the items he has entered, but not remove them. Allowing another user to change items in a calendar will allow him to change all items in the calendar, but not to remove any item.

13.5 AS/400 Exchanging Distributions with Remote Systems

OfficeVision/400 communications is based on SNA, and the security measures discussed earlier in this publication for SNADS are also valid for OfficeVision/400. Refer to Chapter 10, "Communications Security in SNA" on page 10-1 for information on SNADS. Setting up AS/400 for communications with a PROFS VM system is discussed in the redbook *VM-AS/400 Connectivity and Functional Use.* Planning for AS/400 in a network and setting up remote users are discussed in *Planning for and Setting Up OfficeVision/400*. These topics are not discussed further in this publication.

The following features will be discussed:

- Send/Receive Messages
- Send/Receive Notes
- Send/Receive Documents

Some general points should be noted:

- To be able to send distributions to users on other systems, the user and the system must be defined in the System Distribution Directory. Avoid the use of National Special Characters in the user ID or the system name, as they might be displayed differently on systems in the network using a different National Language support.
- The system will not be able to detect duplicate user IDs on remote systems. User ID and address should establish a unique identification for users on other systems. Using the system name as the address and the user profile as the user ID will make the name and address unique in a network.
- Distribution lists on remote systems can be included in distribution lists on the local system. If a remote distribution list, included on a distribution list on the local system, points back to the local system, distributions sent to the remote distribution list will not be retransmitted to the local system.
- The use of default user IDs and default user addresses is discussed in the manual *Setting Up and Planning for OfficeVision/400*.
- OfficeVision/400 users can exchange final-form documents, revisable documents, and messages with PROFS user on the VM/370 system.
- OfficeVision/400 users can exchange notes or messages with non-PROFS users on a PROFS system.
- OfficeVision/400 users can exchange final-form documents, revisable documents, and messages with S/36 Personal Service users. Distributions exchanged with S/36 Personal Service can be made personal.
- Exchanging distributions with users on other systems may cause different translation of National Special Characters.
- User IDs must conform to naming conventions on all systems in the network.
- When the communication facilities are set up to exchange distributions with remote systems, the AS/400 is prepared to convert incoming distributions to a format known to the system.

Table 13-3 on page 13-35 and Table 13-4 on page 13-35 show the conversion of distributions from S/36 and PROFS to AS/400 format.

S/36	**AS/400**
Document RFTS36	Document RFTDCA
Document FFTS36	Document FFTDCA
Note FFTS36	Note FFTDCA
Message	Message

Table 13-3. Document Format Conversion Between S/36 and AS/400. When exchanging distributions with S/36, the distributions are automatically changed to a format known to AS/400.

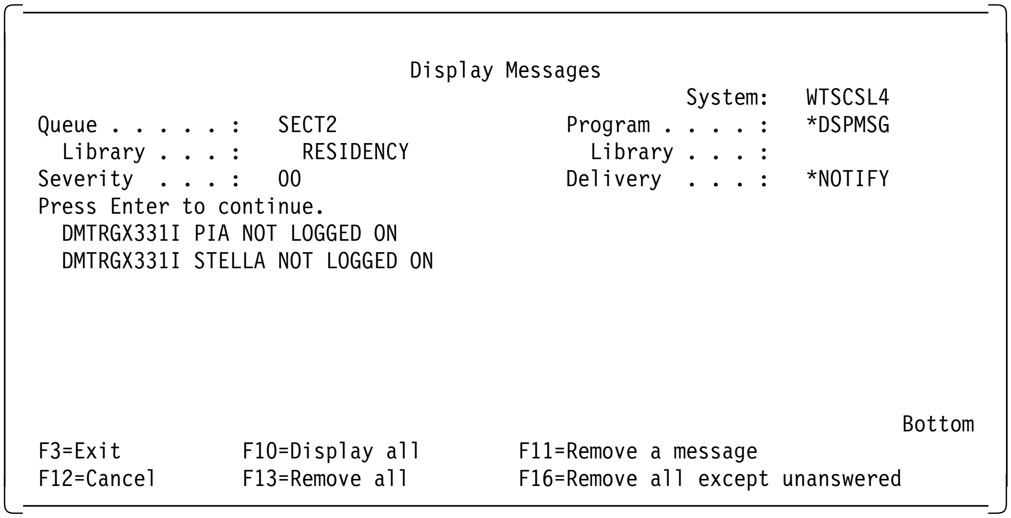
The PROFS format RFT-D cannot be sent to an AS/400. PROFS will return a error, saying that the document must be finalized before sending.

PROFS	**AS/400**
Document RFT-D	Not supported
Document RFT-F	Document RFTDCA
Note 1403W6	Note FFTAS400
Message	Message Queue

Table 13-4. Document Format Conversion Between PROFS and AS/400. Exchanging distributions with a PROFS system the distribution formats are changed automatically to a format know by the AS/400.

13.5.1 Sending Messages

The user ID and the address of the receiver of the note can be keyed, or distribution lists can be used. Using distribution lists will help prevent keying errors or using incorrect addressee information.



```
                             Display Messages
                                                       System:   WTSCSL4
 Queue . . . . . :   SECT2                  Program . . . . :   *DSPMSG
   Library . . . :     RESIDENCY              Library . . . :
 Severity  . . . :   00                     Delivery  . . . :   *NOTIFY
 Press Enter to continue.
   DMTRGX331I PIA NOT LOGGED ON
   DMTRGX331I STELLA NOT LOGGED ON




                                                                    Bottom
 F3=Exit          F10=Display all       F11=Remove a message
 F12=Cancel       F13=Remove all        F16=Remove all except unanswered
```

Figure 13-23. Display Messages Screen with Network Message. An example of message returned from VM to the AS/400 in response to a message sent to users on the VM system. The message is generated by VM and cannot be modified on the AS/400 to present more relevant information

In the example in Figure 13-23, a message has been sent to two User IDs on a VM system. Messages will be not be kept to be displayed when users log on, as they are kept in the AS/400 message queue.

In this example, PIA is a non-existent user ID and STELLA is not signed on, but the sender cannot see the difference. Using distribution lists will help prevent this confusion.

If the receiver of the message is not signed onto the system, the message shown in Figure 13-23 on page 13-35 will be returned to the message queue of the sender.

13.5.2 Receive Messages

When a message is sent from PROFS, it will be sent to the user′s message queue on the AS/400. PROFS does not support personal messages. When a message is sent from Personal Services/36, the message will be placed on the recipient′s mail log. If the message was made personal on the S/36, it will be treated as personal on the AS/400.

13.5.3 Sending Notes

The user should use distribution lists for the reasons stated above.

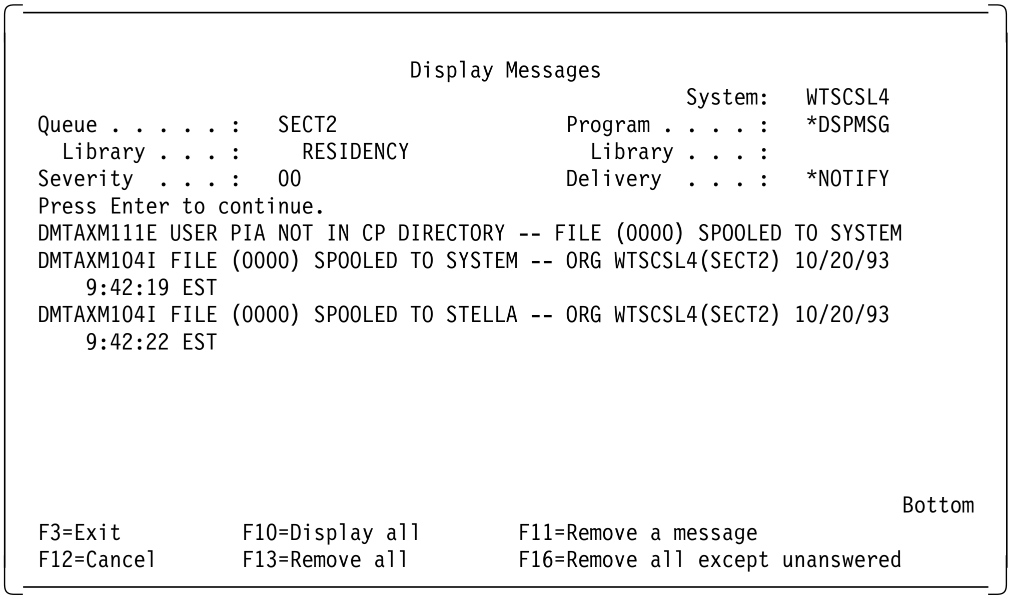
Sending a note requesting confirmation of delivery on the AS/400 will cause SNADS to inform the sender that the note was delivered and later inform the sender when the note was viewed. This is an AS/400 feature that does not apply to notes sent to VM systems. Requesting confirmation of delivery of a note sent to PROFS will cause SNADS to confirm delivery of the note to the system, and not to the user on that system.

If the network cannot deliver the note to the recipient, a message will be returned and placed in the sender′s mailbox.

When a user sends a note to a VM ID , confirmation that the note is spooled to the recipient will be placed in the sender′s message queue.

If the recipient of a note is unknown to the receiving system, a message will be returned to the sender. The returned message will arrive in the user′s message queue, not on the mail log.

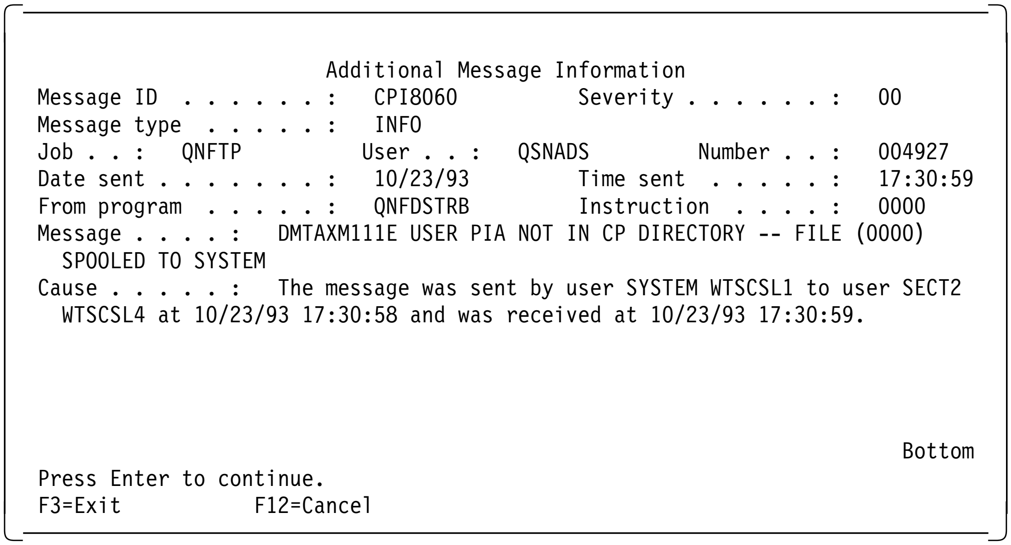
An example of the messages is included below.



```
                              Display Messages
                                                       System:   WTSCSL4
Queue . . . . . :   SECT2                Program . . . . :   *DSPMSG
  Library . . . :     RESIDENCY            Library . . . :
Severity  . . . :   00                   Delivery  . . . :   *NOTIFY
Press Enter to continue.
DMTAXM111E USER PIA NOT IN CP DIRECTORY -- FILE (0000) SPOOLED TO SYSTEM
DMTAXM104I FILE (0000) SPOOLED TO SYSTEM -- ORG WTSCSL4(SECT2) 10/20/93
    9:42:19 EST
DMTAXM104I FILE (0000) SPOOLED TO STELLA -- ORG WTSCSL4(SECT2) 10/20/93
    9:42:22 EST










                                                                        Bottom
F3=Exit          F10=Display all        F11=Remove a message
F12=Cancel       F13=Remove all         F16=Remove all except unanswered
```

Figure 13-24. Display Messages Screen with Network Confirmation. Sending a note to a non-existent user in the network will cause the first two messages to appear on the sender's message queue. If the user exists on the remote system, the last message on the queue will be returned.

As the messages sent to the user's message queue are generated at the VM system and cannot be modified on the AS/400, they will seem unfamiliar to the user. Pressing Help will provide the user with further information about what went wrong with the distribution. The Help display gives the name of the remote system and the non-existent user ID. This should be enough to make the user able to identify the problem, whether it is a typing error or an error in the distribution list. If the error cannot be found in the user's environment, he must contact the system operator to find out if there is an incorrect entry in the System Distribution Directory.



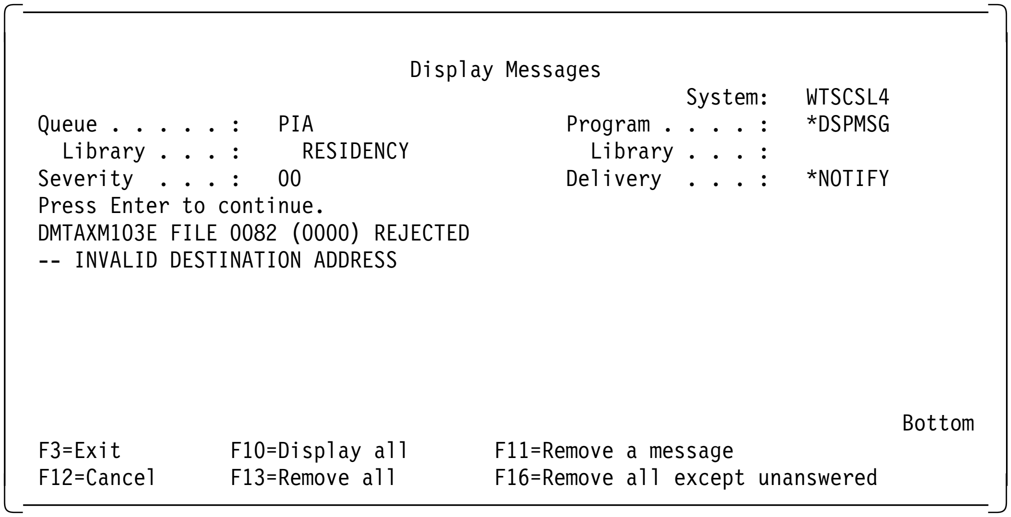
```
                        Additional Message Information
Message ID  . . . . . . :   CPI8060         Severity . . . . . . :   00
Message type  . . . . . :   INFO
Job . . :   QNFTP          User . . :   QSNADS        Number . . :   004927
Date sent . . . . . . . :   10/23/93        Time sent  . . . . . :   17:30:59
From program  . . . . . :   QNFDSTRB        Instruction  . . . . :   0000
Message . . . . :   DMTAXM111E USER PIA NOT IN CP DIRECTORY -- FILE (0000)
  SPOOLED TO SYSTEM
Cause . . . . . :   The message was sent by user SYSTEM WTSCSL1 to user SECT2
  WTSCSL4 at 10/23/93 17:30:58 and was received at 10/23/93 17:30:59.





                                                                        Bottom
Press Enter to continue.
F3=Exit           F12=Cancel
```

Figure 13-25. Additional Message Information for Network Confirmation. The HELP display for the message in Figure 13-23 on page 13-35. It gives the receiver of the message the system name that generated the message and the time the message was generated.

13.5.4 Receive Notes

A note can be answered or keyed using PROFS, and will be directed to the mail log of the AS/400 recipient. PROFS does not enable users to work on behalf of other users, and a PROFS note or a PROFS document cannot be made personal. This must be taken into consideration if the AS/400 is receiving sensitive material from PROFS.

When a note is received from a PROFS user, it is possible to use the REPLY function. The received note will show the address of RSCS instead of the system name and the user must correct it. If the user does not remember to change the address before replying to the note, the message shown in Figure 13-26 will be returned to the user's message queue.



```
                              Display Messages
                                                     System:   WTSCSL4
Queue . . . . . :   PIA                       Program . . . . :   *DSPMSG
  Library . . . :     RESIDENCY                 Library . . . :
Severity  . . . :   00                        Delivery  . . . :   *NOTIFY
Press Enter to continue.
DMTAXM103E FILE 0082 (0000) REJECTED
-- INVALID DESTINATION ADDRESS





                                                                     Bottom
F3=Exit        F10=Display all      F11=Remove a message
F12=Cancel     F13=Remove all       F16=Remove all except unanswered
```

Figure 13-26. Display Messages Screen with Network Rejection

S/36 Personal Services follows the same guidelines as AS/400 regarding personal distributions and personal mail. It is also possible to have users do work on behalf of other users. Indirect users have the same options to have their personal mail printed or not.

Notes created on the S/36 can be made personal and will be treated as personal notes on the AS/400.

Notes created on the S/36 and sent to AS/400 can be answered and returned without losing the mark for personal distribution.

13.5.5 Sending Documents

The user should use distribution lists for the reasons stated above. A document sent to a PROFS user will be transformed to a PROFS file automatically, and all information contained in a memo-slip will be deleted. Errors regarding the distribution will be documented as shown in figure Figure 13-24 on page 13-37.

The PROFS system does not support the function to have users work on another user's behalf, and therefore the personal mark cannot be placed on documents. This means that a PROFS document can be viewed by any user authorized to the mail log containing the PROFS document.

The PROFS document is automatically transformed into a format that can be understood at the AS/400 and be revised by the AS/400 text editor.

13.6 Conclusion on Security in OfficeVision/400

Office is an IBM delivered application that can be secured like any other application. Normal AS/400 security applies to Office, with a few differences caused by the nature of the document library services security. The differences are:

- Access codes:
  - Are **not recommended** as a means of securing documents.
  - Should only be used if documents are migrated from another system that uses access codes.
- Personal documents:
  - Are recommended for extra protection of sensitive documents, when users are working on behalf of others.

OfficeVision/400 requires an Office administrator for enrollment purposes. Great consideration should be given to the number of users and the status of the users who are granted the authority of Office administrator. This Office administrator has access to change certain items on the user profiles in the system, which are unrelated to Office activities.

It is possible to enroll administrators who do not have authority to system objects. See *Planning for and Setting Up OfficeVision/400* for details.

Office users can be limited to working with Office only through the LMTCPB(*YES) on the user profile, limiting access to commands in the system and commands used in documents. If the users have a need to use other applications on the system, they should be limited through the use of user written menus.

It is not possible to exclude users with USRCLS(*SECOFR) from all access to Document Library Objects. This fact could be used to make users with USRCLS(*SECOFR) Office administrators for enrollment purposes in order to limit the number of users with access to user profiles on the system.

Overall, the Office application can be made secure if the administrators and users have the correct understanding of the environment. Setting up and maintaining an Office environment is a job that should not be given to someone as a secondary responsibility. Neither should the entire responsibility for the Office environment be placed with someone without providing them with the necessary understanding of AS/400 resource security and the implications of their responsibility.

At one extreme, Office applications are considered difficult, messy and out of control. At the other extreme, they may be considered a small application with no system impact. Neither view is accurate. The Office application can be a well structured, well functioning and smoothly running application. However, proper attention must be given to security both during Office implementation and throughout maintenance and administration.

Chapter 14. Auditing the AS/400

IBM has categorized the following security functions according to International Organization for Standardization (ISO) standard 7498-2:

- **Identification and authentication**

  Identifies users to the system and provides proof that they are who they claim to be.

- **Access control**

  Determines which users can access which resources.

- **Data confidentiality**

  Protects an organization′s sensitive data from unauthorized disclosure.

- **Data integrity**

  Ensures that data is in its original form and that it has not been altered.

- **Security management**

  Administers, controls and reviews a business′ security policy.

- **Nonrepudiation**

  Assures that the message was sent by the appropriate individual.

This standard should be implemented in any company′s security policy. As an auditor you should have it mind.

14.1 Security Auditing

Security auditing, in the context used here, refers to two different activities:

1. Day-to-day security monitoring tasks
2. Periodic security reviews and audits

Prior to performing either of these audit tasks, it is important for you to obtain a general understanding of:

- The company business and structure
- The operating environment and the policies and procedures of the Information Systems department

The day-to-day monitoring tasks should be part of the security administration. Full-time security administrators will probably perform these tasks every day, while part-time administrators may select to do the monitoring tasks on a less frequent basis.

Periodic reviews and audits may be performed by internal or external auditors. Depending on the size and security needs of an organization, periodic reviews and audits may be performed annually or less frequently. The purpose of this chapter is to discuss the necessity of these activities and how to carry them out rather than giving guidelines for their frequency.

14.2 Auditing Environment

Security auditing involves using various AS/400 commands to access the system history log, the journal information, system values, network attributes, user profiles and authorization lists.

Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Others require *AUDIT special authority.

To facilitate the auditing process, it is easier to use a software tool to automate the gathering of auditing information. This way, you don′t have to execute all of the suggested CL commands interactively, just create a CL program with several of these steps and execute the compiled CL program in batch.

The QUSRTOOL library, supplied with OS/400, contains a variety of commands and programs that can be used for the auditing purpose.

Note: QUSRTOOL contains programming source code and is provided on an ″as is″ basis. See Appendix I, “Security Tools in Library QUSRTOOL” on page I-1 for more information.

There is a lot of information within an AS/400 that could be checked in a periodic audit, but to what extent depends on how much time you can spend. There is a tool, Security/400, that will produce some of the reports you may need, but you should be prepared to make your own programs too if you want to be thorough. You will have to use a combination of CL, a high-level language, Query or SQL.

The Security/400 can be very useful in a day-to-day audit. It has reports for every entry logged in the audit journal. Refer to Chapter 15, “IBM Security/400” on page 15-1 for more information about Security/400, and to the *AS/400 Security Reference* for information about QAUDJRN, the audit journal.

14.3 Gaining a General Understanding of the Company

The knowledge of a company′s business, structure, policies and procedures is an important prerequisite to perform the security auditing for several reasons:

- By knowing what the company services are and where it performs its operations, you can better understand the structure of the applications being processed, the configurations of communications and devices, and the descriptions of subsystems, libraries, and other objects.
- The organization chart can provide you with the information of how the company has established (or you should establish) its User Profiles, Group Profiles, and Authorization Lists.
- An understanding of IS department policies and procedures provides you with the company′s methodology for establishing security. Company policies serve as a starting point for you to determine if appropriate procedures are in place and to develop the specific auditing steps to address these procedures.

Company financial reports, hardware and software listings, policy and procedure manuals, and the organization charts all serve as the useful references in the initial information gathering.

14.4 Audit and Control Concepts

Effective controls reduce business risk and the amount of substantive auditing test work required.

When you want to audit an AS/400 there are a few things that must be done before you start on the system itself. The AS/400 does not live a life by its own. It is part of an organization, and must be run according to this organization′s needs.

14.4.1 Audit Objectives

Several types of auditors exist (internal/external, Financial/Data Processing), but the common audit objectives are to:

- Assess the completeness, accuracy, existence and integrity of the company data
- Assess the adequacy of the control procedures
- Ensure the compliance with the regulatory and legal requirements
- Ensure the compliance with the company policies
- Enhance the operational efficiency

To meet the audit objectives, you should completely understand the control environment. This complete understanding can assist you to assess the level of risk.

14.4.2 What is Included in the Audit

It is important that you, the auditor, and the client agree upon what the audit should include, such as:

- Buildings, doors, windows, and so on.
- The physical access to terminals/printers/telecommunications equipment
- OfficeVision/400
- All applications, or a selection

14.4.3 Guidelines for Auditing

The audit should be conducted according to the company′s written security plan. If such a plan does not exist, ask for a copy of the ″Plan to protect the business processes″. (Many call this a ″disaster plan″, which does not always give the correct associations). The ″Plan to protect the business processes″ will give you an idea of application ownership and the responsibilities within the company, and what is important to them. If neither of these plans exist you and the client must work something out.

14.4.4 Control Concepts

In order to protect the business information, the management establishes and implements a system of control procedures. These control procedures should:

- Safeguard the company assets from unauthorized and invalid use and modification
- Protect the confidentiality of the company data
- Provide the adequate segregation of duties
- Ensure the system and data availability

When assessing these control procedures, the auditors use a variety of test methodologies. The three primary test methodologies are:

- Inquiry
- Observation
- Penetration

When a computer is used to store and process the corporate information, assessing the software security controls become an essential part of any audit. Global security options and specific access rules define the logical controls that protect the business information.

Understanding, evaluating, and testing the control procedures are the primary audit objectives. When the logical security control is the method used to control access to the business information, it is often easier and more efficient to use *penetration* as the test methodology. Additionally, you can rely more on the test results obtained from penetration than those obtained from inquiry or observation.

14.4.5 The Flow of Information

A company can not function without established routines. Some of these routines influence how security is handled, in general and in the AS/400. Many routines may seem bureaucratic, but they found the base the security must be built upon.

Register a New User

Some sort of routine must be followed when a new user profile is created. The person who registers new users should receive a form that contains at least the following:

- The name of the user
- The user class
- Any deviation from the default values in the CRTUSRPRF command, verified by the person responsible for the AS/400 security
- Authority to the applications, verified by the application owners

After the form has been registered someone should verify that the user profile and the authorizations are registered correctly, and then file the form.

Register a User Who Leaves

Some sort of routine must be followed when a user leaves the company, or gets a leave of absence. A form should be filled out and given to the person responsible for AS/400 security.

Before deleting a user profile, are the following must be checked:

- If the user has programs that adopts his authority
- If the user owns other objects: if so, who decides if they are to be deleted, or transferred to a new owner

Note: Try to avoid transferring objects to the QDFTOWN profile. It may be better to create one or more user profiles with PASSWORD=*NONE and STATUS=*DISABLED as temporary owners until a new owner is found. The new owner may not always be obvious to the person who deletes the user profiles. Such temporary owners are better than QDFTOWN and may be used if there is a security requirement to delete a user profile immediately. You may of course set PASSWORD=*NONE and STATUS=*DISABLED in the user profile to be deleted until the ownership is sorted out. The main thing is that there is a working routine.

14.4.6 Applications and Ownership

Some applications may be considered more important than others. As an auditor you must know:

- Which applications should be secured.
- Which applications (if any) must not be secured.
- Which applications are continuously being changed, and which are ″frozen″.
- Which libraries are included in an application.
- Who within the company are the owners of the different applications.
- Which user profiles own the objects within an application. Ownership is extremely important and plays a key role in a secure system.
- Who can request changes to an application, and how are these request for changes documented and carried out.

Note: QUSRTOOL has a program, CHGLIBOWN, that can change the owner of a library, and all the objects within the library. This is a very powerful tool. Please don′t use it until:

- The owner of the application agrees that a change should be made. The owner may have detailed knowledge and must always be consulted.
- You have changed the owner of the library manually and see that it works. Whenever you make a change of ownership, you must be able to reverse the process if something does not work properly afterwards.
- You know if the library contains programs that adopt their owner′s authority.

14.4.7 Program Maintenance and Development

Programs should be developed and maintained in a test environment. No programmers should be given access to the production environment. Everything they need should be copied to this environment for them.

Program development and maintenance should be done according to written documentation.

Programmers should document what they have done. When the programs are tested and found to work as intended, the source must be backed up and the programs moved to the production environment.

One way to do that is to:

- Move the source programs to a temporary library in the production environment and compile the programs to a temporary library.
- Compare the source listings with the programmer′s documentation. This is to ensure that nobody has placed a logical ″bomb″, a Trojan horse, in a program, compiled it and removed the ″bomb″.

- When the programs are found to be okay, they can be put to use.

Todays programmers are often in complete control of a system, and nobody seems to care.

14.4.8 The Source Files

The content of the source files are very valuable to the company. Many have invested heavily in developing/tailoring their applications to meet their business needs. The applications must be protected from unauthorized changes. Only a very limited number of users should have access to the source files.

An application should have its own library for source files. If source files are found in other libraries, such as QGPL, the content must be verified.

14.4.9 Philosophy

In order to reduce security exposure and business risk, the access rules should be based upon the principle of the least necessary privilege. That is, the individuals should be given only the access authority that they require to perform their jobs.

From a control perspective, it is important to:

- Protect the integrity of the operating system
- Implement adequate controls to limit access to the data and programs
- Log, monitor, handle and report the security related events
- Limit access to the access control facilities

The implementation of security should conform to the defined security policies. Specific AS/400 facilities provide a method of enforcing the defined security policies.

> **IMPORTANT!**
>
> The responsibility of administering the security environment must be assigned at the organizational level that will allow the adequate resolution of the security concerns. You must monitor the implemented controls routinely because controls tend to deteriorate over time.

14.4.10 Audit Strategies

To assess the implemented controls, you may use several audit strategies. The primary strategies are:

- Status monitoring
- Event monitoring
- Rule analysis

Status Monitoring: Status monitoring involves displaying, listing, and reporting the system-level security related parameters, and allows you to evaluate these system security related parameters.

Event Monitoring: Event monitoring usually takes place at the system and application level. You can use two approaches while monitoring events. One approach involves comparing the values collected at two different times and evaluating changes. The other approach involves recording events as they occur and reporting the events. The primary steps of establishing the event reporting are:

1. Establish the event recording mechanisms
2. Identify the important security related events
3. Record or log the important security related events
4. Extract and report the events

You can report the important security related events with two distinct methods. First, the security events can be organized, formatted, and summarized. In this method, you report the security related events just after the event has occurred. A discussion on how to establish the event reporting mechanisms in the AS/400 environment is presented in Section 6.4, "Using the Audit Journal to Report on System Activity" on page 6-4.

The second method involves using a mechanism to notify the individuals of the events as they occur in real time. Usually a message is sent to the master console operator or to the security administrator. Although this facility is not included within OS/400, the capabilities exist to establish this type of real time reporting mechanism.

Rule Analysis: Rule analysis is based upon the same theory as status monitoring. That is, access control rules are captured at some instant for analysis. The difference between the two strategies is that status monitoring focuses on the system-level rules and parameters, while rule analysis focuses on the individual access rules as they apply to the users and resources.

14.5 AS/400 Security Controls

AS/400 security controls can be divided into physical security control and logical security control.

The physical security control is a very important part of security auditing, as it helps ensure the availability and reliability of the entire system. Several physical security control issues are as applicable to the AS/400 as they are to any central processor.

We will discuss the physical security control in the following order:

- Machine room
- UPS (Uninterruptable Power Supply)
- AS/400 keylock
- Workstation/terminal
- Backup tapes and documentation

The logical security control can establish the baseline of controls that ensure system integrity, so as to protect the system resources and enforce the security policies.

We will discuss the logical security control in terms of:

- Security-related system values and network attributes
- User profiles and group profiles
- Authorization lists
- Audit journal

14.5.1 Physical Security

Don′t ignore the physical security control! Before processing the logical security control on AS/400 system, you should confirm physical security control has been implemented well.

Machine Room

The machine room should be water-proofed and fire-proofed. The door should be locked to control the entrance. Only authorized personnel should be able to gain access to the machine room. Each entrance should be logged for preventing unauthorized access.

UPS (Uninterruptable Power Supply)

You should determine whether or not the company has a UPS system. If it does, you should check UPS-related controls to ensure the UPS allows for a normal shutdown in case of a power outage.

AS/400 Keylock

The physical position of the keylock on the AS/400 controls the use of the specific AS/400 facilities. The keylock should be set to the *Secure* position and then removed. The key should be stored at a secure location. Physical security control should limit the use of the key to changing the lock-switch position. For more information about the key position and the associated functions, refer to the *Application System/400: System Operator′s Guide*.

Workstation/Terminal

Company policy should prohibit recording the confidential information (for example, signon, and other activities that involve password entry) on workstation/terminal record/play keys. You should perform a spot check of workstations/terminals to assure the compliance with these policies. If there is a key for keyboard lock at the workstation/terminal, ensure the keyboard is locked and the key is removed when the workstation/terminal is inactive.

Backup Tapes

Consult the ″Plan to protect the business processes″ to see how the backup routines, including labeling and storing of the tapes, are performed. Verify that these routines are followed, and check if anyone is able to steal, duplicate or borrow a tape without being noticed.

Industrial espionage is a threat that must not be overlooked. Is there a security exposure in this area?

14.5.2 Logical Security Controls

Security-Related System Values and Network Attributes

In the AS/400 system, there are many system values and network attributes which affect system-wide operations. Some of them are important to security. They can be identified by the WRKSYSVAL (Work with System Value) command. See Chapter 2, “System Values and Network Attributes” on page 2-1 for more information.

User Profiles and Group Profiles

The user profile has several roles on the system:

- It contains the security-related information that controls how the user signs on the system, what the user is allowed to do after signing on, and how the user′s actions are audited.
- It contains the information that is designed to customize the system and adapt it to the user.
- It is a management and recovery tool for the operating system. The user profile contains the information about the objects owned by the user and all the private authorities to objects.
- The user profile name identifies the user′s jobs and printer output.

If the security level (QSECURITY) system value on your system is 10, the system automatically creates a user profile when someone signs on with a user profile that does not already exist on the system.

If the QSECURITY system value on your system is 20 or higher, a user profile must exist before a user can sign on.

A group profile is a special type of user profile. It serves two purposes on the system:

Security tool
A group profile provides a method for organizing the authorities on your system and sharing them among the users. You can define the object authorities for group profiles rather than for each individual user profile.

Customizing tool
A group profile can be used as a pattern for creating individual user profiles. Most people who are part of the same group have the same customizing needs, such as the initial menu and the default printer. You can define these things in the group profile and then copy the group profile to create individual user profiles.

You create group profiles in the same way that you create individual user profiles. The system recognizes a group profile when you add the first member to it. At that point, the system sets the information in the profile indicating that it is a group profile.

See Chapter 3, “User Profiles and Group Profiles” on page 3-1 for more information.

Authorization Lists

You can group the objects with similar security requirements using an authorization list. An authorization list contains a list of users and the authority that the users have to the objects secured by the list. Each user can have a different authority to the set of objects the list secures.

You can also use an authorization list to define the public authority for the objects on the list. If the public authority for an object is set to *AUTL, the object gets its public authority from its authorization list.

The authorization list object is used as a management tool by the system. It contains a list of all objects which are secured by the authorization list. This information is used to build displays for viewing or editing the authorization list objects.

See Chapter 5, “Authorization Lists” on page 5-1 for more information.

Audit Journal

The security audit journal is the primary source of auditing information on the system. You can use the auditing function provided by the system to gather the information about security-related events that occur on the system.

You can define auditing on your system at three different levels:

- System-wide auditing that occurs for all users
- Auditing that occurs for the specific objects
- Auditing that occurs for the specific users

You use system values, user profile parameters, and object parameters to define auditing.

When a security-related event that may be audited occurs, the system checks whether you have selected that event for audit. If you have, the system writes a journal entry in the current receiver for the security auditing journal (QAUDJRN in library QSYS).

See Chapter 6, “Audit Journal” on page 6-1 for more information.

14.6 Using OS/400 Facilities to Audit

For security to be effective, the security controls must be monitored regularly. The commands and utilities on the OS/400 automate the auditing and security management. Additionally, the auditor product IBM Security/400 (5764-006) can be used to produce the reports to assist in this task. You can see Chapter 15, “IBM Security/400” on page 15-1 for more information.

14.6.1 Day-to-Day Monitoring

Security, once established at the desired level, tends to deteriorate over time. The reasons for this effect lie in the dynamics of the computer use and the complexity of the environment. The typical factors are:

- New objects created by system users
- New users enrolled on the system
- Changes of object ownership - authorization not adjusted
- Changes of responsibilities - user group changed
- Temporary authorizations - not revoked
- New products installed
- Maintenance applied - security level lowered and not reset, and so on

It is therefore necessary that the key security controls should be monitored on a regular basis in the following two categories:

1. Reviewing the key security events

2. Checking the status of the key security controls

The best way to keep an eye on what is happening on the system is to use the audit journal (QAUDJRN). Many types of events, such as security violations, changes to user profiles, work management, and network attributes, are logged in the journal receiver. You can create your own programs or queries to produce reports, or you could order them from menus in Security/400.

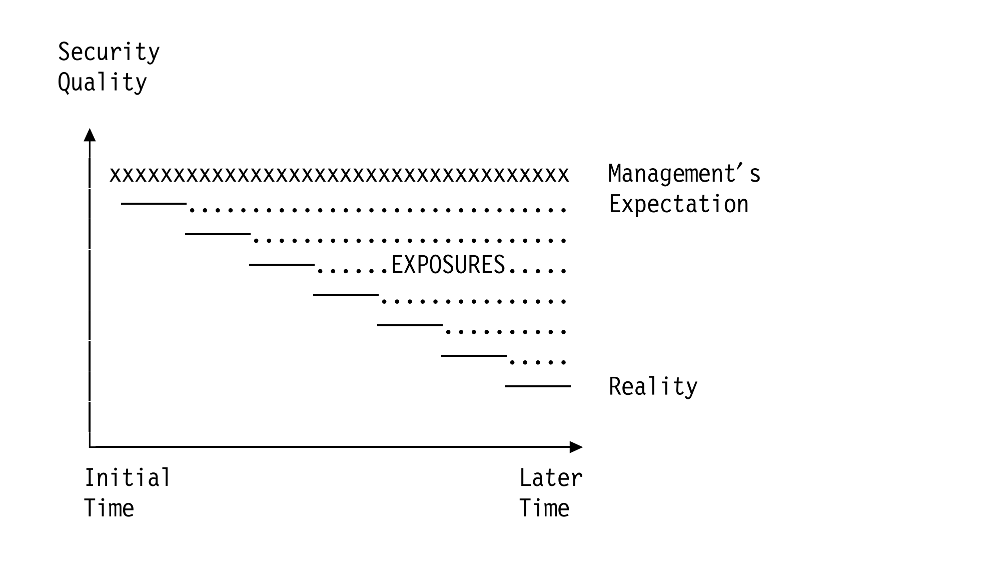


Figure 14-1. Effective Security Level. The effective security level of most systems tends to deteriorate over time. This can be counteracted by good security monitoring facilities, reasonable care by the security officer, and periodic security audits. The point is that security requires a continuing effort.

Status Monitoring

The primary day-to-day activity to maintain good security is an analysis of security events. Complementary to this is monitoring the key security controls, which includes

- Global controls and options at the system level
- Related user definitions and privileges
- Related object definitions and authorizations

Global Controls and Options at the System Level: The considerations described under AS/400 security controls for physical security and security-related system values/network attributes may be applied by the security administrator on a more frequent (although not daily) basis.

Critical User Profiles: Critical User Profiles should be checked regularly, such as profiles with special authorities and IBM-supplied user profiles where the default passwords are published.

Privileged Profiles: All User Profiles with special authorities such as *ALLOBJ, *SECADM and *AUDIT should be extracted and compared with an authorized list of such users. The analysis should include other properties like PASSWORD(*NONE).

The DSPAUTUSR command will print the following information for all User Profiles:

- User Profile name
- Group Profile name
- Date password was last changed
- An indicator if the password is *NONE
- Description text

To do this, enter:

```
DSPAUTUSR OUTPUT(*PRINT)
```

To print other User Profile information, enter:

```
DSPUSRPRF USRPRF(User Profile) TYPE(*ALL) OUTPUT(*PRINT)
```

Along with the basic profile information, this prints all commands, devices, and objects that the user has specific authority on, objects the user owns, and group members (if the profile is a Group Profile).

IBM-Supplied User Profiles: IBM-supplied user profiles should be checked in the following ways:

- For the user profiles designed as object owners or batch processing, you should verify that their password are all *NONE to prevent them being used to sign on to the system.
- for the user profiles shipped with default passwords, you should verify that these passwords cannot be used to sign on. The default passwords should be changed immediately after installing the system. In addition, they should be changed periodically (in case they become known, are reset to the defaults, and so on). But you still should verify that the rest of parameters except the password have not been changed. See 3.3, “IBM-Supplied User Profiles” on page 3-7 for more information.

Critical Objects: For critical objects, the public and specific authority should be checked to see that it meets the security guidelines or objectives. Table 14-1 shows some of the critical system objects.

Table 14-1. List of Critical System Objects

Object Name	Object Type	Recommended Public Authority
QSYS	*LIB	*USE
QUSRSYS	*LIB	*USE
QHLPSYS	*LIB	*USE
QGPL	*LIB	*CHANGE
QDOC	*LIB	*USE
QBASE	*SBSD	*USE
QCTL	*SBSD	*USE
QBATCH	*SBSD	*USE
QINTER	*SBSD	*USE
QCMN	*SBSD	*USE

Critical installation objects are the production libraries containing programs, source programs, and files for applications with high protection requirements (for confidentiality or integrity reasons). They should be added to the above list.

Event Monitoring

The journal files and history log contain, among other information, the security-related events that must be monitored. It is necessary that this information be extracted and documented in security reports for management review. We suggest the following priorities:

1. Analyze the reported changes to security definitions and rules.

2. Analyze the access granted to highly critical objects.
3. Analyze the attempted violations.

Access to Critical Objects: The security rules and definitions tend to become too generous over time. A periodic review of all rules (for an application) is the best way to correct this situation. While this approach may be acceptable for the majority of objects in an installation, exposures may not be tolerable for a subset of highly critical objects. For these, you should verify the rules and monitor the access granted more frequently.

When security is implemented in an organized way, the critical system and user objects can be easily identified. A list of these objects can be used to select and document all access to these objects from the log files. The unexpected access granted may be an indication of incorrect security rules and definitions.

The access authority (but not an access log) to a specific object can be printed with the following command:

```
DSPOBJAUT OBJ(library/object) OBJTYPE(type) OUTPUT(*PRINT)
```

For the program GRPPRFR1 in library SECURITY, you would enter:

```
DSPOBJAUT OBJ(SECURITY/GRPPRFR1) OBJTYPE(*PGM) OUTPUT(*PRINT)
```

For the users in the authorization list (if one exists for the object), you can use the following command:

```
DSPAUTL AUTL(AUTL1) OUTPUT(*PRINT)
```

where AUTL1 is the name of the authorization list.

OS/400 records the events such as incorrect passwords, attempting to access an object with insufficient authority, and so forth. These events are recorded in QHST. Examples of the specific commands for listing the elements of QHST are shown later in this chapter.

In addition, all changes and accesses to physical files can be recorded in journals. While the use of journals is more related to application design than overall *system* security, you will need to understand their use within a given installation.

A journal can include:

- The identification of the job, user, and time of access
- The before- and after- images of all file changes
- The records of when the file was opened, closed, and saved

In addition, journals can be used to record other activities. For example, a validity checking program for the Change User Profile command could be set up to record all uses of the command. The CP entry in the QAUDJRN records all changes in a user profile.

A journal entry cannot be altered by any user, even the security officer. A complete journal can be deleted, but this is easily detected.

14.6.2 Periodic Reviews

Periodic reviews may be performed at various levels of detail. A ″diagnostic″ review might be limited to answering global questionnaires. More detailed reviews would analyze the system status, verify the security definitions, or include a statistical analysis of the security definitions and, if appropriate, program code review.

The following sections discuss typical audit programs at different levels of details in the following areas:

- Physical security
- System status and options
- User and group definition and maintenance
- Access authorization

Physical Security

You should check physical security periodically according to the information introduced in Section 14.5.1, “Physical Security” on page 14-8.

System Status and Options

The system aspect of the review focuses on global questions, the system status, and possible extensions and modifications.

Security-Related System Values: Security-related system values (for example, QSECURITY, QMAXSIGN, and so on) should be reviewed to see that effective global security values have been established.

The Work with System Value (WRKSYSVAL) command allows the user to display a list of system values. These can be shown on the screen or sent to a printer. Depending on the value selected in the SYSVAL parameter, either all system values (value of *ALL) or specific groups (for example, value of *SEC) can be displayed or printed. The command example shown below produces the report in Figure 14-2 on page 14-15 that shows security related system values.

```
WRKSYSVAL SYSVAL(*SEC) OUTPUT(*PRINT)
```

```
                                             System Values                                                 Page     1
 5738SS1 V2R3M0  930411                                                                        RCHASM01  04/30/93  15:09:59
                Current                        Shipped
  Name          value                          value                          Description
  QALWUSRDMN    *ALL                           *ALL                           Allow user domain objects in libraries
  QAUDCTL    >  *OBJAUD                        *NONE                          Auditing control
                *AUDLVL                        ' '
  QAUDENDACN    *NOTIFY                        *NOTIFY                        Auditing end action
  QAUDFRCLVL    *SYS                           *SYS                           Force auditing data
  QAUDLVL    >  *AUTFAIL                       *NONE                          Security auditing level
                *SECURITY                      ' '
                *SAVRST                        ' '
                *DELETE                        ' '
                *SYSMGT                        ' '
  QCRTAUT       *CHANGE                        *CHANGE                        Create default public authority
  QCRTOBJAUD    *NONE                          *NONE                          Create object auditing
  QDSPSGNINF >  1                              0                              Signon display information control
  QINACTITV     *NONE                          *NONE                          Inactive job time-out
  QINACTMSGQ    *ENDJOB                        *ENDJOB                        Inactive job message queue
  QLMTDEVSSN    0                              0                              Limit device sessions
  QLMTSECOFR >  0                              1                              Limit security officer device access
  QMAXSGNACN >  2                              3                              Action to take for failed signon attempts
  QMAXSIGN   >  3                              15                             Maximum signon attempts allowed
  QPWDEXPITV    *NOMAX                         *NOMAX                         Password expiration interval
  QPWDLMTAJC    0                              0                              Limit adjacent digits in password
  QPWDLMTCHR    *NONE                          *NONE                          Limit characters in password
  QPWDLMTREP    0                              0                              Limit repeating characters in password
  QPWDMAXLEN    10                             10                             Maximum password length
  QPWDMINLEN    1                              1                              Minimum password length
  QPWDPOSDIF    0                              0                              Limit password character positions
  QPWDRQDDGT    0                              0                              Require digit in password
  QPWDRQDDIF    0                              0                              Duplicate password control
  QPWDVLDPGM    *NONE                          *NONE                          Password validation program
  QRMTSIGN   >  *VERIFY                        *FRCSIGNON                     Remote signon control
  QSECURITY  >  30                             10                             System security level
      Note:  > means current value is different from the shipped value
                                       * * * * *   E N D  O F  L I S T I N G   * * * * *
```

Figure 14-2. Report of All Security-Related System Values

The report displays the current values, the values when the system was shipped and brief descriptions of each security related system value. This report is useful for monitoring the setting of the security values on the system.

See Section 2.9, “System Values and Network Attributes Recommendations” on page 2-14 for more information when inspecting the security-related system values.

Modifications and Extensions: When the validity checkers, validation programs (for example, the password validation program) or exits in any code are used, a review of the source code used to create the program or exit may be necessary.

Security-Related Network Attributes: Network attributes control how your system communicates with other systems. Some network attributes control how remote requests to process jobs and access information are handled. Three security-related network attributes directly affect security on your system and should be reviewed:

Job action (JOBACN)
PC Support access (PCSACC)
Distributed data management (DDMACC)

To review the value of a network attribute, use the Display Network Attribute (DSPNETA) command. To set the value of a network attribute, use the Change Network Attribute (CHGNETA) command.

See Section 2.9, “System Values and Network Attributes Recommendations” on page 2-14 for more information when inspecting the security-related network attributes.

User and Group Definition and Maintenance

The Display User Profile (DSPUSRPRF) command shows the content of a user profile. The output from this command can be used to produce reports that can assist in auditing application-level controls on the AS/400 system. The key parameters for this command are:

- USRPRF: The user profile name or *ALL (to display all user profiles)
- TYPE: The type of information to be displayed, which can be *ALL, *BASIC, *CMDAUT, *DEVAUT, *OBJAUT, *OBJOWN, or *GRPMBR. Refer to the *Application System/400 Programming: Control Language Reference* for the complete descriptions of these options.
- OUTPUT: The output of the command may be presented on the display (*), sent to the printer (*PRINT), or sent to a database file (*OUTFILE).
- OUTFILE: The database file to which the details are sent if the *OUTFILE option is chosen in the OUTPUT parameter.

The most appropriate for auditing purposes is to display all users to an output file with *BASIC detail level. The command would be entered as follows:

```
DSPUSRPRF USRPRF(*ALL) TYPE(*BASIC) OUTPUT(*OUTFILE) OUTFILE(filename)
```

Model files are provided in the system for the various detail level types, as follows:

- QADSPUPO for TYPE(*OBJOWN)
- QADSPUPA for TYPE(*OBJAUT)
- QADSPUPB for TYPE(*BASIC)

Note: TYPE(*BASIC) can capture the information of *ALL profiles. The other types capture information for one user profile at a time.

Using the command in the example above enables the basic information on all user profiles to be sent to a database file. Query/400 or user written programs can access this data to produce audit reports. This output could also be used as input for subsequent display commands.

Database files can be used as any of the following:

- Input to Query/400
- Input to user written programs
- Input to subsequent commands
- A combination of the above

to produce the audit reports shown in the following sections.

Group Profiles: Group profiles are used to organize the users by job function. This organization simplifies the assignment and administration of object authorities. By authorizing the users through a smaller number of entries, the administration is simplified.

A member of a group profile can share all authorities granted to the group. Authorities given specifically to the member override the authority of the group profile. A user profile can be associated with one and only one group profile.

To meet an audit requirement of showing the group profiles on the system, first create an outfile using the following command:

```
DSPUSRPRF USRPRF(*ALL) TYPE(*BASIC) OUTPUT(*OUTFILE) OUTFILE(filename)
```

The OUTFILE created is then used as input to a RPG program, see Figure 14-3, to produce the group profile audit report in Figure 14-4 on page 14-19.

For the purpose of this example, the outfile created is USRPRF in library SECURITY. A logical file, USRPRFL1, based on the outfile USRPRF, keyed by the GROUP PROFILE field UPGRPF, must also be created.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
    1       **************************************************************************
    2       *                                                                        *
    3       *    GRPPRFR1 - This program reads the outfile created by the            *
    4       *               DSPUSRPRF *ALL *BASIC command, finds the user            *
    5       *               profile that is a group profile and uses this            *
    6       *               user profile to read a logical file that is              *
    7       *               keyed by the GROUP PROFILE field to obtain all           *
    8       *               users belonging to that group.                           *
    9       *                                                                        *
   10       **************************************************************************
  100      FUSRPRF  IF  E                    DISK
  101      F*                                      Outfile of DSPUSRPRF command
  200      FUSRPRFL1IF  E           K        DISK
  201      F*                                      Logical File on USRPRF keyed by
  202      F*                                      GROUP PROFILE field
  300      F            QSYDSUPB                          KRENAMEGRPLFREC
  400      FPRTFILE O   F     132     OA     PRINTER
  401      F*                                      Report file
  500      IUPSPAU      DS
  600      I                                        1  10 AUTH1
  700      I                                       11  20 AUTH2
  800      I                                       21  30 AUTH3
  900      I                                       31  40 AUTH4
 1000      I                                       41  50 AUTH5
 1100      I                                       51  60 AUTH6
 1200      I                                       61  70 AUTH7
 1300      I                                       71  80 AUTH8
 1400      I                                       81  90 AUTH9
 1500      I                                       91 100 AUTH10
 1600      I                                      101 110 AUTH11
 1700      I                                      111 120 AUTH12
 1800      I                                      121 130 AUTH13
 1900      I                                      131 140 AUTH14
 2000      I                                      141 150 AUTH15
```

Figure 14-3 (Part 1 of 3). RPG Program to Report Special Authorities of Group Profiles

```
2100     C                    Z-ADD60        LNCNT   30
2200     C                    READ USRPRF                   10
2300     C           *IN10    DOWEQ'0'
2400     C           UPGRPI   IFEQ '*YES'
2500     C*                                        Check GROUP PROFILE indicator
2600     C           LNCNT    IFGT 53
2700     C                    EXCPTHEAD1
2800     C                    EXCPTHEAD2
2900     C                    EXCPTHEAD3
3000     C                    Z-ADD8         LNCNT
3100     C                    END
3200     C                    EXCPTPRT01
3300     C                    ADD  3         LNCNT
3400     C*                                        If it is a GROUP PROFILE use
3500     C*                                        it as a key to obtain all
3600     C*                                        USER PROFILES for the GROUP.
3700     C                    MOVELUPUPRF    GRPPF  10
3800     C                    EXSR GROUP
3900     C                    END
4000     C                    READ USRPRF                   10
4100     C                    END
4200     C                    SETON                         LR
4300     C           GROUP    BEGSR
4400     C           GRPPF    SETLLUSRPRFL1                 11
4500     C           GRPPF    READEUSRPRFL1                 12
4600     C           *IN12    DOWEQ'0'
4700     C                    EXCPTPRT02
4800     C                    ADD  1         LNCNT
4900     C           GRPPF    READEUSRPRFL1                 12
5000     C                    END
5100     C                    ENDSR
5200     OPRTFILE E    3          HEAD1
5300     O                                58 'GROUP  PROFILES'
5400     O                                80 'AND  RELATED  USERS'
5500     OPRTFILE E 2             HEAD2
5600     O                                08 'USER'
5700     O                                18 'USER'
5800     O                                44 'GROUP'
5900     O                                56 'GROUP'
6000     OPRTFILE E 1             HEAD3
6100     O                                 9 'PROFILE'
6200     O                                18 'CLASS'
6300     O                                31 'OWNER'
6400     O                                46 'PROFILE'
6500     O                                54 'IND'
6600     O                                77 'SPECIAL AUTHORITIES'
```

Figure 14-3 (Part 2 of 3). RPG Program to Report Special Authorities of Group Profiles

```
    6700      OPRTFILE E 3              PRT01
    6800      0                         UPUPRF    10
    6900      0                         UPUSCL    23
    7000      0                         UPOWNR    35
    7100      0                         UPGRPF    48
    7200      0                         UPGRPI    55
    7300      0                         AUTH1     68
    7400      0                         AUTH2     79
    7500      0                         AUTH3     90
    7600      0                         AUTH4    101
    7700      0                         AUTH5    112
    7800      0                         AUTH6    121
    7900      0                         AUTH7    130
    8000      OPRTFILE E 1              PRT02
    8100      0                         UPUPRF    12
    8200      0                         UPUSCL    23
    8300      0                         UPOWNR    35
    8400      0                         UPGRPF    48
    8500      0                         UPGRPI    55
    8600      0                         AUTH1     68
    8700      0                         AUTH2     79
    8800      0                         AUTH3     90
    8900      0                         AUTH4    101
    9000      0                         AUTH5    112
    9100      0                         AUTH6    121
    9200      0                         AUTH7    130
                                   * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 14-3 (Part 3 of 3). RPG Program to Report Special Authorities of Group Profiles

```
                                         GROUP  PROFILES   AND  RELATED  USERS

     USER      USER                    GROUP       GROUP
   PROFILE    CLASS        OWNER        PROFILE      IND    SPECIAL AUTHORITIES

 GROUP        *USER       *USRPRF      *NONE        *YES   *NONE
   TEST       *USER       *GRPPRF      GROUP        *NO    *NONE

 PGMRUSER     *PGMR       *USRPRF      *NONE        *YES   *JOBCTL    *SAVSYS
   BAKER      *PGMR       *USRPRF      PGMRUSER     *NO    *JOBCTL    *SAVSYS
   ITSCID021  *SECOFR     *USRPRF      PGMRUSER     *NO    *JOBCTL    *SAVSYS
   JENS       *SECOFR     *USRPRF      PGMRUSER     *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   MYHRA      *PGMR       *USRPRF      PGMRUSER     *NO    *JOBCTL    *SAVSYS

 SECOFRUSER   *SECOFR     *USRPRF      *NONE        *YES   *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   CRAIG      *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID01   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID02   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID03   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID05   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID10   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL
   ITSCID20   *SECOFR     *USRPRF      SECOFRUSER   *NO    *ALLOBJ    *AUDIT     *JOBCTL    *SAVSYS    *SECADM  *SERVICE *SPLCTL

 WOEUSER      *USER       *USRPRF      *NONE        *YES   *NONE
   WOEGROUP   *USER       *USRPRF      WOEUSER      *NO    *NONE
```

Figure 14-4. Special Authorities of Group Profiles and their Associated Users

Object Authorities: To evaluate the users-to-objects relationships, use the value *OBJAUT on the TYPE parameter of the DSPUSRPRF command. TYPE, however, does not allow the user profile parameter to be stated as *ALL. The information, therefore has to be gathered in two steps:

1. Use the command to gather *ALL profiles with the type *BASIC into an OUTFILE. Use this OUTFILE as input to step 2.

2. Read each profile from the OUTFILE and, using the DSPUSRPRF command again, send the type *OBJAUT to a second OUTFILE.

Run a query against the second OUTFILE produced to present the information as required.

The program shown in Figure 14-5 on page 14-21 can be used to gather the information (both steps mentioned above are run within the control language (CL) program).

Note: This program runs for a long time and should be submitted as a batch job.

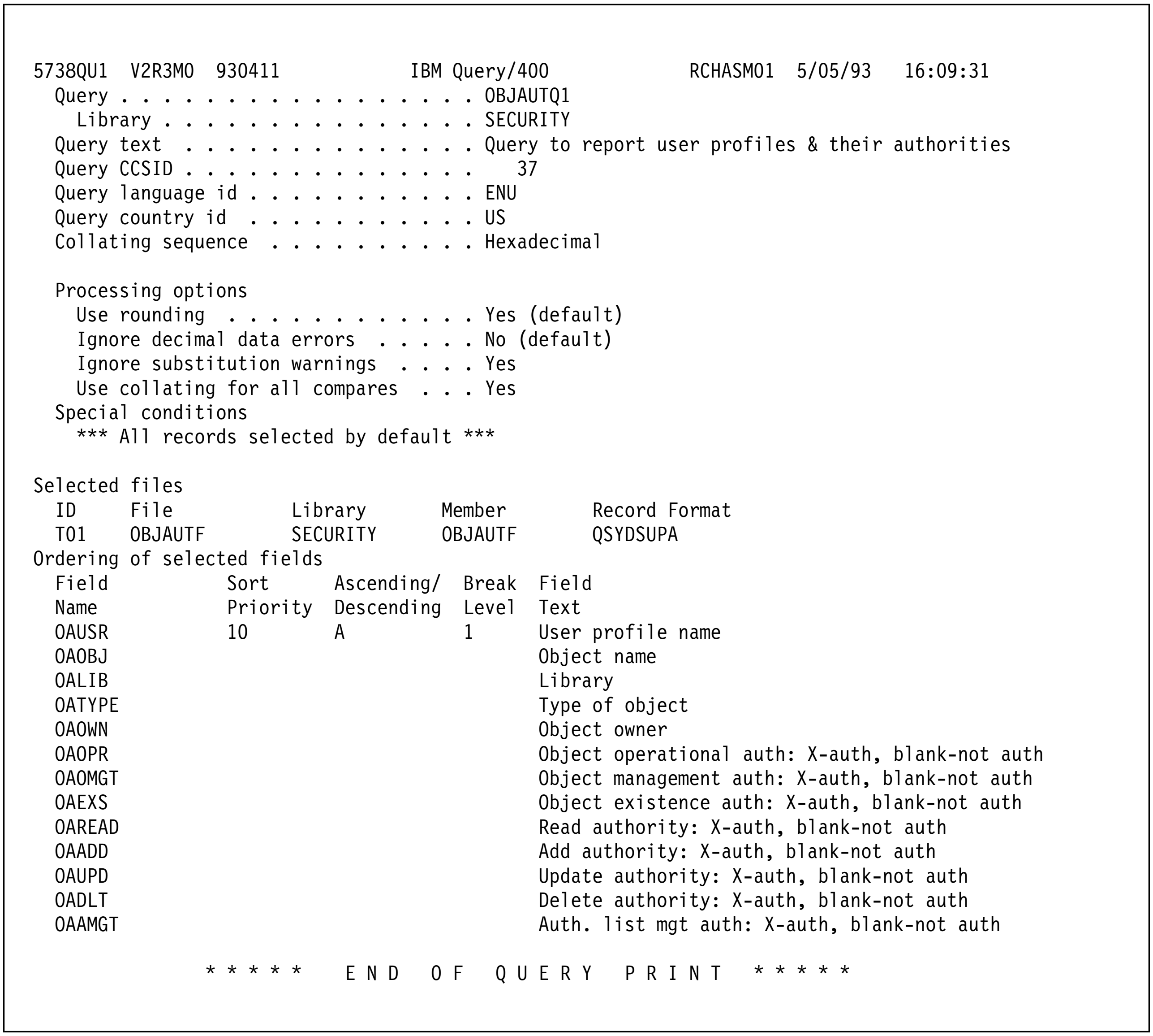
```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
     1            PGM
     2 /*************************************************************************/
     3 /*  OBJAUTCL: This is a sample program to show all users on a system        */
     4 /*            and the objects they are authorized to. The *OBJAUT parm       */
     5 /*            cannot be done on *ALL user profiles. Thus, the *BASIC         */
     6 /*            profile for all users is displayed to a database file.         */
     7 /*            Each of the records relates to an individual user profile.     */
     8 /*            Each user profile is then displayed with the *OBJAUT           */
     9 /*            parameter to another database file with the *ADD option.       */
    10 /*            This file is then available for AS/400 queries to be run       */
    11 /*            against it.                                                    */
    12 /*                                                                           */
    13 /*************************************************************************/
   102            DCLF       FILE(QADSPUPB) /* Format of the Display User +
   103                                        Profile database file */
   200            DSPUSRPRF  USRPRF(*ALL) TYPE(*BASIC) OUTPUT(*OUTFILE) +
   300                         OUTFILE(SECURITY/USRPRF) /* Display *ALL +
   301                         User Profile */
   400            OVRDBF     FILE(QADSPUPB) TOFILE(SECURITY/USRPRF) /* +
   401                         Override System supplied format file to +
   402                         new file created */
   403            DLTF       FILE(SECURITY/OBJAUTF) /* Delete file if it +
   404                         already exists */
   405            MONMSG     MSGID(CPF2105) /* Ignore if not found */
   500  READ:     RCVF       /* Read each record on the file created */
   501            MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
   600            DSPUSRPRF  USRPRF(&UPUPRF) TYPE(*OBJAUT) +
   700                         OUTPUT(*OUTFILE) +
   800                         OUTFILE(SECURITY/OBJAUTF) OUTMBR(*FIRST +
   801                         *ADD) /* Display *OBJAUT for each user +
   802                         profile read */
   900            GOTO       CMDLBL(READ)
  1000  EOF:      ENDPGM
                             * * * *  E N D  O F  S O U R C E  * * * *
```

*Figure 14-5. Control Language Program to Produce *ALL Profile OUTFILE*

From this outfile (OBJAUTF), a query (shown in Figure 14-6 on page 14-22) can report on all users on the system and their object authorities as shown in Figure 14-7 on page 14-23.

This report (Figure 14-7 on page 14-23) shows the users on the system, the objects they are authorized to, the type of authorization they have, and the type of object. You could investigate the necessity of authorization as reported and recommend updates where necessary.

In order to determine the object-to-user access, the outfile produced to evaluate the user-to-object access is used. The query definition must be changed to sort the records in object sequence before the report is printed.



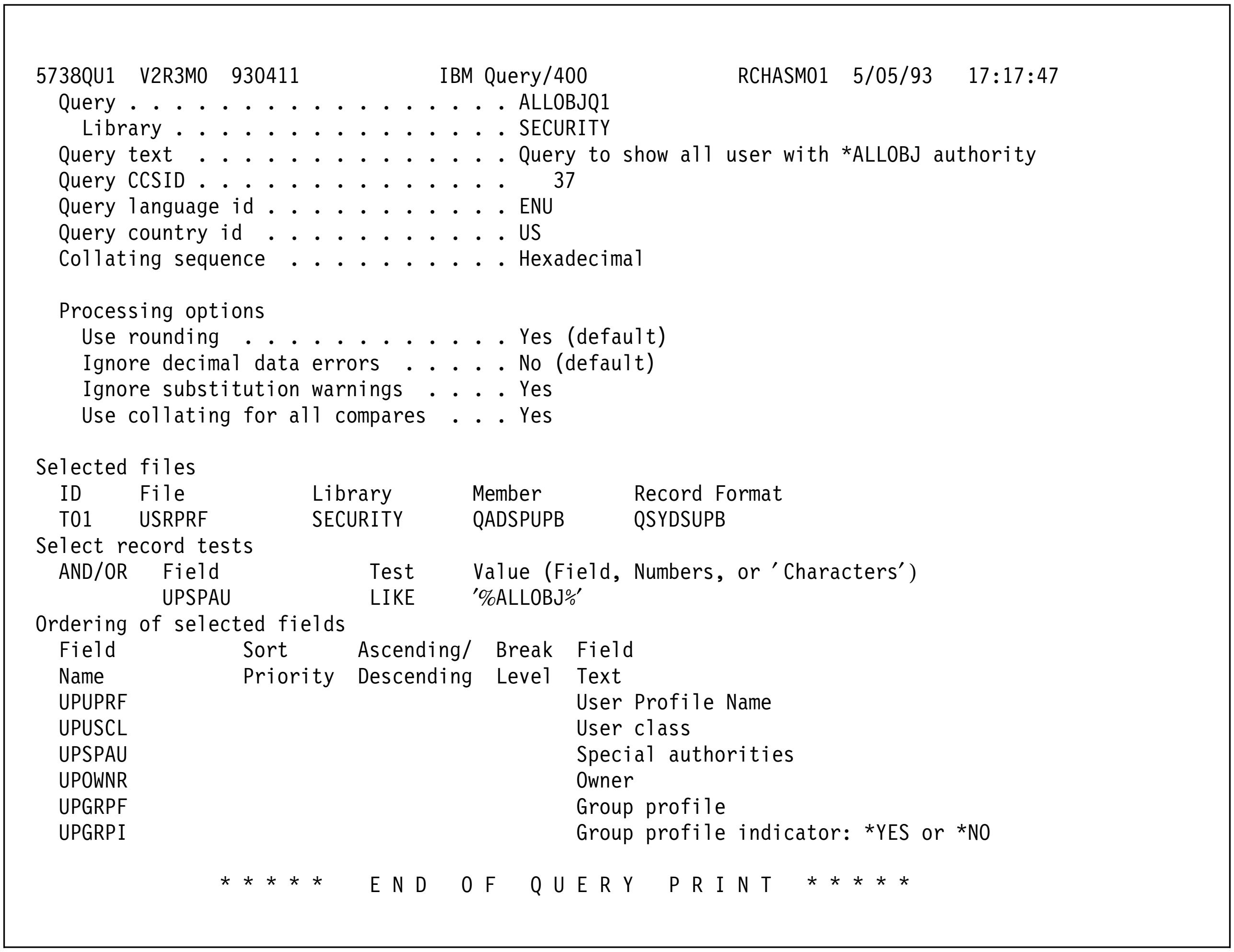
```
5738QU1  V2R3M0  930411           IBM Query/400           RCHASM01  5/05/93   16:09:31
  Query . . . . . . . . . . . . . . . . . OBJAUTQ1
    Library . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . . Query to report user profiles & their authorities
  Query CCSID . . . . . . . . . . . . . .    37
  Query language id . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . US
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes
  Special conditions
    *** All records selected by default ***

Selected files
  ID     File           Library       Member        Record Format
  T01    OBJAUTF        SECURITY      OBJAUTF       QSYDSUPA
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  OAUSR           10        A           1      User profile name
  OAOBJ                                        Object name
  OALIB                                        Library
  OATYPE                                       Type of object
  OAOWN                                        Object owner
  OAOPR                                        Object operational auth: X-auth, blank-not auth
  OAOMGT                                       Object management auth: X-auth, blank-not auth
  OAEXS                                        Object existence auth: X-auth, blank-not auth
  OAREAD                                       Read authority: X-auth, blank-not auth
  OAADD                                        Add authority: X-auth, blank-not auth
  OAUPD                                        Update authority: X-auth, blank-not auth
  OADLT                                        Delete authority: X-auth, blank-not auth
  OAAMGT                                       Auth. list mgt auth: X-auth, blank-not auth

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 14-6. Query Definition to Report on User Profiles and Their Authorities

```
                                       User Profiles on System and Their Object Authorities

USER        OBJECT      LIBRARY     TYPE      OWNER       OBJECT       OBJECT      OBJECT     READ  ADD  UPDATE  DELETE  AUTL
                                                          OPERATIONAL  MANAGEMENT  EXISTENCE                             MANAGEMENT

CRAIG       CRAIG       QSYS        *USRPRF   QSECOFR          X           X                   X     X     X       X
            QSECOFR     QSYS        *USRPRF   QSYS             X           X                   X     X     X       X
            ITSCID05    QUSRSYS     *MSGQ     ITSCID05         X           X           X       X     X     X       X

EVANS       QDEVELOP    QSYS        *LIB      QSECOFR          X                               X
            EVANS       QSYS        *USRPRF   MAB              X           X                   X     X     X       X
            STRDBG      QSYS        *CMD      QSYS             X                               X
            TRCINT      QSYS        *CMD      QSYS             X                               X
            TAADSPADP   QSYS        *AUTL     XVT1925          X                               X     X     X       X
            GLOOP       QUSRSYS     *MSGQ     GLOOP            X           X           X       X     X     X       X
            TROND       QUSRSYS     *MSGQ     TROND            X           X           X       X     X     X       X
            WOEGROUP    QUSRSYS     *MSGQ     WOEGROUP         X           X           X       X     X     X       X
            WOEUSER     QUSRSYS     *MSGQ     WOEUSER          X           X           X       X     X     X       X

SECURITY    SECURITY    QSYS        *USRPRF   STAGG            X           X                   X     X     X       X
            QSECOFR     QSYS        *USRPRF   QSYS             X           X                   X     X     X       X
* * *  E N D  O F  R E P O R T  * * *
```

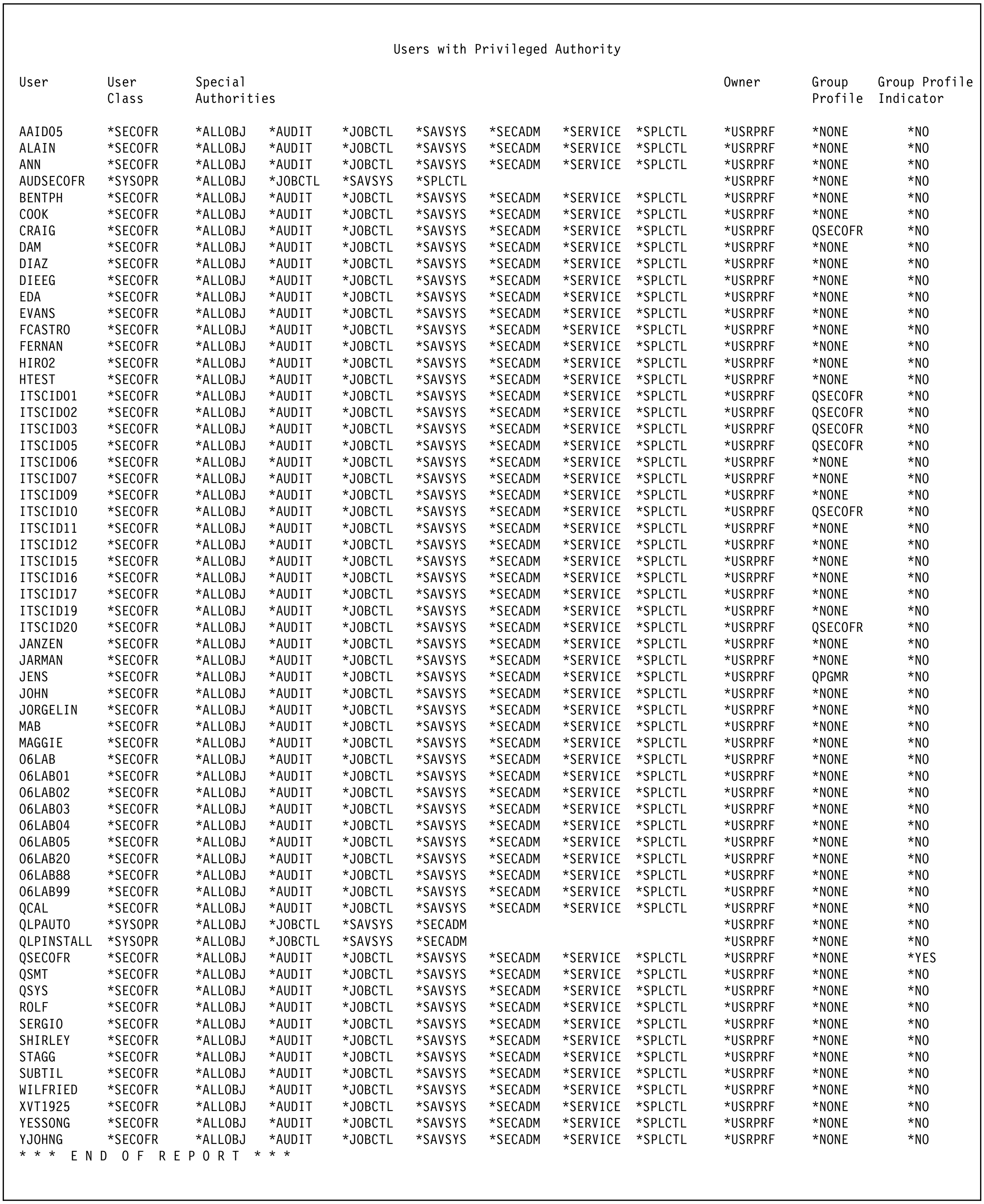
Figure 14-7. Report on User Profiles and Their Authorities

Privileged Authorities: As part of the auditing function, the users that are given *ALLOBJ authority should be monitored. To list these users, the OUTFILE produced by the DSPUSRPRF command with the value *BASIC in the TYPE parameter is used. The query definition for this requirement is shown in Figure 14-8 on page 14-24 and the formatted report is shown in Figure 14-9 on page 14-25. The users that have *ALLOBJ authority can be monitored, and the adequate control can be applied.



```
5738QU1  V2R3M0  930411           IBM Query/400            RCHASM01  5/05/93   17:17:47
  Query . . . . . . . . . . . . . . . . . . ALLOBJQ1
    Library . . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . . . Query to show all user with *ALLOBJ authority
  Query CCSID . . . . . . . . . . . . . . .    37
  Query language id . . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . . US
  Collating sequence  . . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . . No (default)
    Ignore substitution warnings  . . . . . Yes
    Use collating for all compares  . . . . Yes

Selected files
  ID     File           Library       Member        Record Format
  T01    USRPRF         SECURITY      QADSPUPB      QSYDSUPB
Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           UPSPAU           LIKE     '%ALLOBJ%'
Ordering of selected fields
  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text
  UPUPRF                                      User Profile Name
  UPUSCL                                      User class
  UPSPAU                                      Special authorities
  UPOWNR                                      Owner
  UPGRPF                                      Group profile
  UPGRPI                                      Group profile indicator: *YES or *NO

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 14-8. Query Definition Showing all Privileged User Profiles



```
                                                   Users with Privileged Authority

User        User        Special                                                                 Owner       Group    Group Profile
            Class       Authorities                                                                         Profile  Indicator

AAID05      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ALAIN       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ANN         *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
AUDSECOFR   *SYSOPR     *ALLOBJ   *JOBCTL   *SAVSYS   *SPLCTL                                   *USRPRF     *NONE        *NO
BENTPH      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
COOK        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
CRAIG       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
DAM         *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
DIAZ        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
DIEEG       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
EDA         *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
EVANS       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
FCASTRO     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
FERNAN      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
HIRO2       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
HTEST       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID01    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
ITSCID02    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
ITSCID03    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
ITSCID05    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
ITSCID06    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID07    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID09    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID10    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
ITSCID11    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID12    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID15    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID16    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID17    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID19    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ITSCID20    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QSECOFR      *NO
JANZEN      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
JARMAN      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
JENS        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     QPGMR        *NO
JOHN        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
JORGELIN    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
MAB         *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
MAGGIE      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB01     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB02     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB03     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB04     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB05     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB20     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB88     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
O6LAB99     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
QCAL        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
QLPAUTO     *SYSOPR     *ALLOBJ   *JOBCTL   *SAVSYS   *SECADM                                   *USRPRF     *NONE        *NO
QLPINSTALL  *SYSOPR     *ALLOBJ   *JOBCTL   *SAVSYS   *SECADM                                   *USRPRF     *NONE        *NO
QSECOFR     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *YES
QSMT        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
QSYS        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
ROLF        *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
SERGIO      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
SHIRLEY     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
STAGG       *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
SUBTIL      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
WILFRIED    *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
XVT1925     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
YESSONG     *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
YJOHNG      *SECOFR     *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL     *USRPRF     *NONE        *NO
* * *  E N D  O F  R E P O R T  * * *
```

Figure 14-9. Report Showing all Privileged User Profiles

Access Authorization

We discuss the access authorization via the following topics:

- Display object authority,
- Display authorization list,
- Adopted authorities,
- Using History Log,
- Auditing successful access to a database file.
- Auditing security configuration changes,

Display Object Authority: The Display Object Authority (DSPOBJAUT) command displays a list of authorized users of an object, and their authorities to the object. This command

```
DSPOBJAUT OBJ(Object-name) LIB(Library-name) OBJTYPE(Object-type) OUTPUT(*)
```

can be executed for only one object at a time. For the specified object the following are shown:

- Object name
- Library that contains the object
- Object owner
- Object type
- Authorization list securing the object
- A list of all the users who are authorized to use the object
- The authorities that each user has for the object

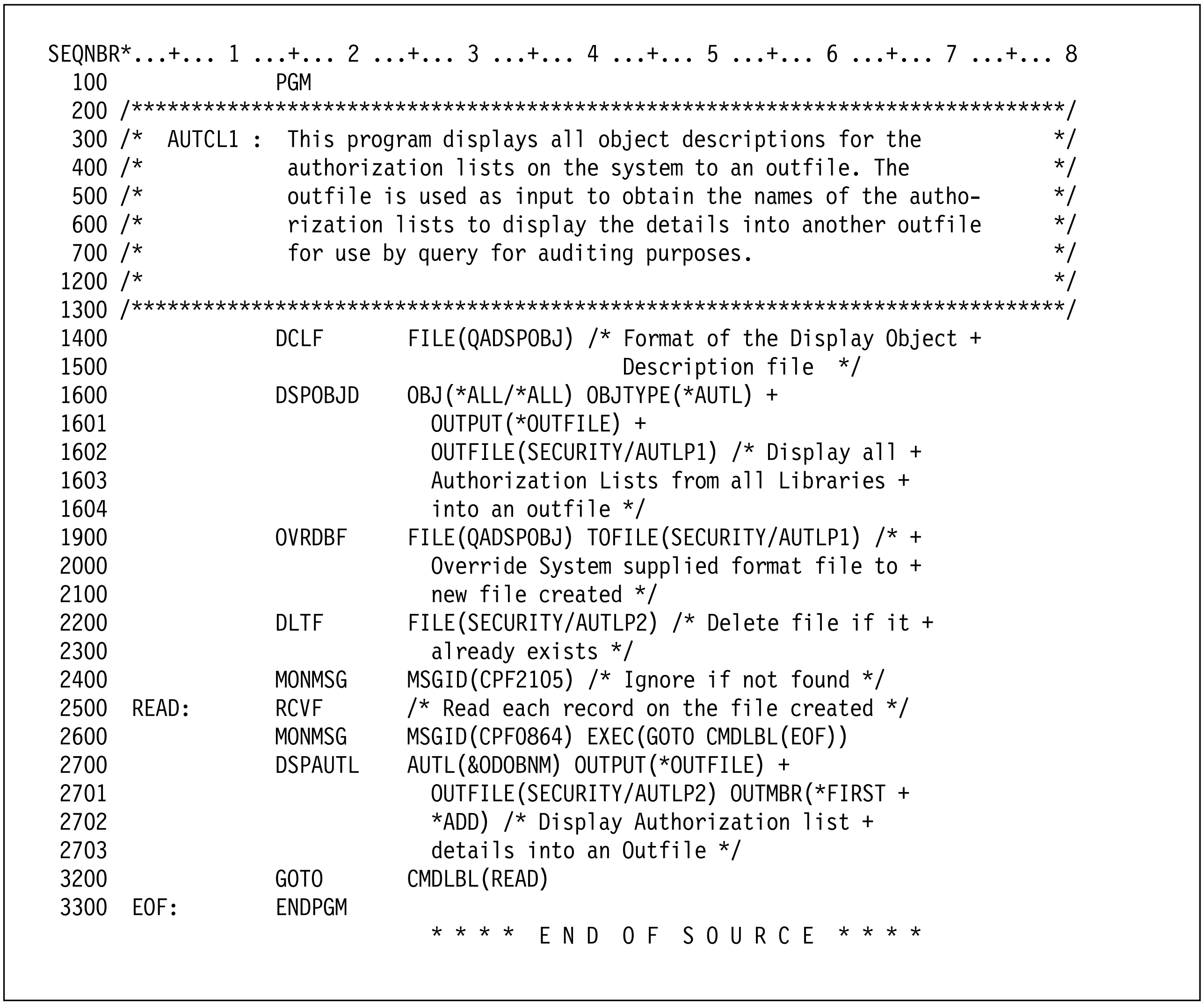
Refer to the *Application System/400 Programming: Control Language Reference* for the detailed definition of the parameters.

For auditing or security management purposes, you might have to list the authorities for a number of critical objects. To create this list all the critical objects should be entered in a file. This file could then be used as the input to a control language program to generate a report that lists all critical objects, their authorized users, and the users′ authorities to those objects.

Display Authorization List: The Display Authorization List (DSPAUTL) command can display, print, or write to a file, the information on one authorization list at a time. You may require a report of all authorization lists on the system, who the owners are, who has authorities to them, and what type of authorities they have. This can be compiled by writing all the authorization lists in the system to a file by using the DSPOBJD command. This file can then be used in a control language program as the input to display the details of each authorization list to a second output file. A query can then be performed to generate a report that shows the users authorized to authorization lists and the authorities that these users have.

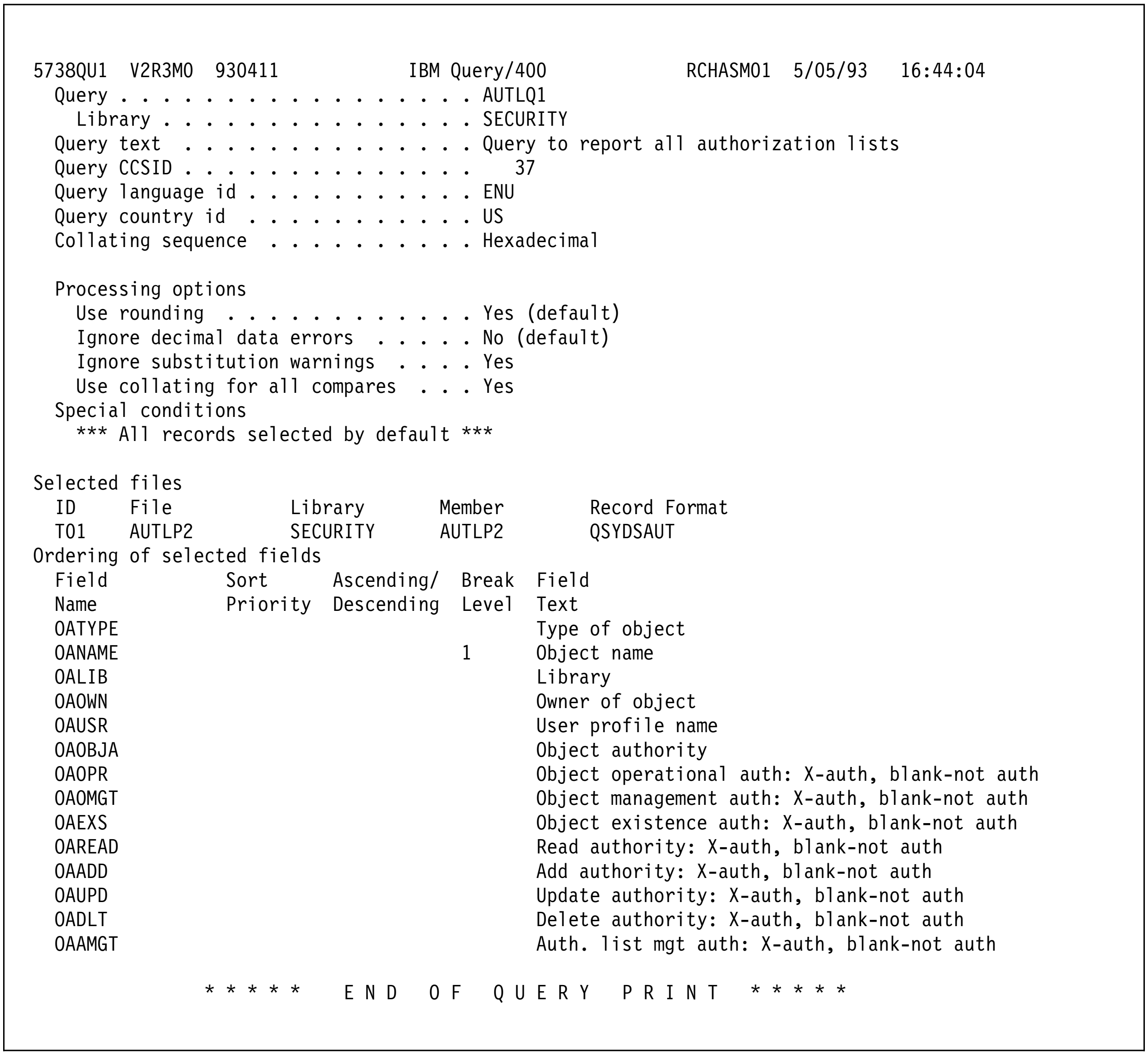
An example of one way to achieve the desired results is shown in Figure 14-10 on page 14-27 (the control language program) and Figure 14-11 on page 14-28 (the query definition). The result report is shown in Figure 14-12 on page 14-29.

The report can further be enhanced to include the list of objects that are secured by all the authorization lists. This additional information can be provided by the Display Authorization List Object (DSPAUTLOBJ) command.



```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
  100             PGM
  200 /******************************************************************************/
  300 /*  AUTCL1 :  This program displays all object descriptions for the          */
  400 /*            authorization lists on the system to an outfile. The           */
  500 /*            outfile is used as input to obtain the names of the autho-      */
  600 /*            rization lists to display the details into another outfile      */
  700 /*            for use by query for auditing purposes.                         */
 1200 /*                                                                            */
 1300 /******************************************************************************/
 1400             DCLF       FILE(QADSPOBJ) /* Format of the Display Object +
 1500                                          Description file  */
 1600             DSPOBJD    OBJ(*ALL/*ALL) OBJTYPE(*AUTL) +
 1601                          OUTPUT(*OUTFILE) +
 1602                          OUTFILE(SECURITY/AUTLP1) /* Display all +
 1603                          Authorization Lists from all Libraries +
 1604                          into an outfile */
 1900             OVRDBF     FILE(QADSPOBJ) TOFILE(SECURITY/AUTLP1) /* +
 2000                          Override System supplied format file to +
 2100                          new file created */
 2200             DLTF       FILE(SECURITY/AUTLP2) /* Delete file if it +
 2300                          already exists */
 2400             MONMSG     MSGID(CPF2105) /* Ignore if not found */
 2500  READ:      RCVF       /* Read each record on the file created */
 2600             MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
 2700             DSPAUTL    AUTL(&ODOBNM) OUTPUT(*OUTFILE) +
 2701                          OUTFILE(SECURITY/AUTLP2) OUTMBR(*FIRST +
 2702                          *ADD) /* Display Authorization list +
 2703                          details into an Outfile */
 3200             GOTO       CMDLBL(READ)
 3300  EOF:       ENDPGM
                          * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 14-10. Program to Show All Authorization Lists in the System



```
5738QU1  V2R3M0  930411           IBM Query/400           RCHASM01  5/05/93   16:44:04
  Query . . . . . . . . . . . . . . . . . AUTLQ1
    Library . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . . Query to report all authorization lists
  Query CCSID . . . . . . . . . . . . . .    37
  Query language id . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . US
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes
  Special conditions
    *** All records selected by default ***

Selected files
  ID     File           Library       Member        Record Format
  T01    AUTLP2         SECURITY      AUTLP2        QSYDSAUT
Ordering of selected fields
  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text
  OATYPE                                      Type of object
  OANAME                               1      Object name
  OALIB                                       Library
  OAOWN                                       Owner of object
  OAUSR                                       User profile name
  OAOBJA                                      Object authority
  OAOPR                                       Object operational auth: X-auth, blank-not auth
  OAOMGT                                      Object management auth: X-auth, blank-not auth
  OAEXS                                       Object existence auth: X-auth, blank-not auth
  OAREAD                                      Read authority: X-auth, blank-not auth
  OAADD                                       Add authority: X-auth, blank-not auth
  OAUPD                                       Update authority: X-auth, blank-not auth
  OADLT                                       Delete authority: X-auth, blank-not auth
  OAAMGT                                      Auth. list mgt auth: X-auth, blank-not auth

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 14-11. Query Definition on All Authorization Lists

```
                                                  Objects on Authorization Lists

TYPE      OBJECT      LIBRARY     OWNER       USER        OBJECT     OBJECT       OBJECT      OBJECT     READ  ADD  UPDATE  DELETE
                                                          AUTHORITY  OPERATIONAL  MANAGEMENT  EXISTENCE

*AUTL     AUTLCHG     QSYS        EVANS       EVANS       *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        EVANS       *PUBLIC     *CHANGE         X                               X     X     X       X

*AUTL     AUTLLESS    QSYS        EVANS       EVANS       *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        EVANS       QUSER       *USE            X                               X
*AUTL                 QSYS        EVANS       *PUBLIC     *CHANGE         X                               X     X     X       X

*AUTL     AUTLMORE    QSYS        EVANS       EVANS       *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        EVANS       QUSER       *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        EVANS       *PUBLIC     *CHANGE         X                               X     X     X       X

*AUTL     JANZEN      QSYS        GROUP       GROUP       USER DEF
*AUTL                 QSYS        GROUP       MAGGIE      *USE            X                               X
*AUTL                 QSYS        GROUP       SERGIO      *CHANGE         X                               X     X     X       X
*AUTL                 QSYS        GROUP       *PUBLIC     *EXCLUDE

*AUTL     QCQRPSAUTL  QSYS        QSYS        QSYS        *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSYS        *PUBLIC     *EXCLUDE

*AUTL     QFMCAUTL    QSYS        QSYS        QSYS        *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSYS        QSECOFR     *CHANGE         X                               X     X     X       X
*AUTL                 QSYS        QSYS        QOSIFS      USER DEF
*AUTL                 QSYS        QSYS        *PUBLIC     *EXCLUDE

*AUTL     QIWSADM     QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSECOFR     *PUBLIC     *USE            X                               X

*AUTL     QRTL        QSYS        QSYS        QSYS        *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSYS        QRTL        *CHANGE         X                               X     X     X       X
*AUTL                 QSYS        QSYS        *PUBLIC     *CHANGE         X                               X     X     X       X
*AUTL     TAACVTQHST  QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X

*AUTL                 QSYS        QSECOFR     *PUBLIC     *EXCLUDE
*AUTL     TAADLTQHST  QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X

*AUTL                 QSYS        QSECOFR     *PUBLIC     *EXCLUDE
*AUTL     TAADSPADP   QSYS        XVT1925     XVT1925     *ALL            X           X           X       X     X     X       X

*AUTL                 QSYS        XVT1925     EVANS       *CHANGE         X                               X     X     X       X
*AUTL                 QSYS        XVT1925     *PUBLIC     *EXCLUDE

*AUTL     TAAENAUSR   QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSECOFR     *PUBLIC     *EXCLUDE

*AUTL     TAARSTALLC  QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSECOFR     *PUBLIC     *EXCLUDE

*AUTL     TAASAVALLC  QSYS        QSECOFR     QSECOFR     *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        QSECOFR     *PUBLIC     *EXCLUDE

*AUTL     WOETEST     QSYS        EVANS       EVANS       *ALL            X           X           X       X     X     X       X
*AUTL                 QSYS        EVANS       QUSER       *CHANGE         X                               X     X     X       X
*AUTL                 QSYS        EVANS       *PUBLIC     *CHANGE         X                               X     X     X       X
* * *  E N D  O F  R E P O R T  * * *
```

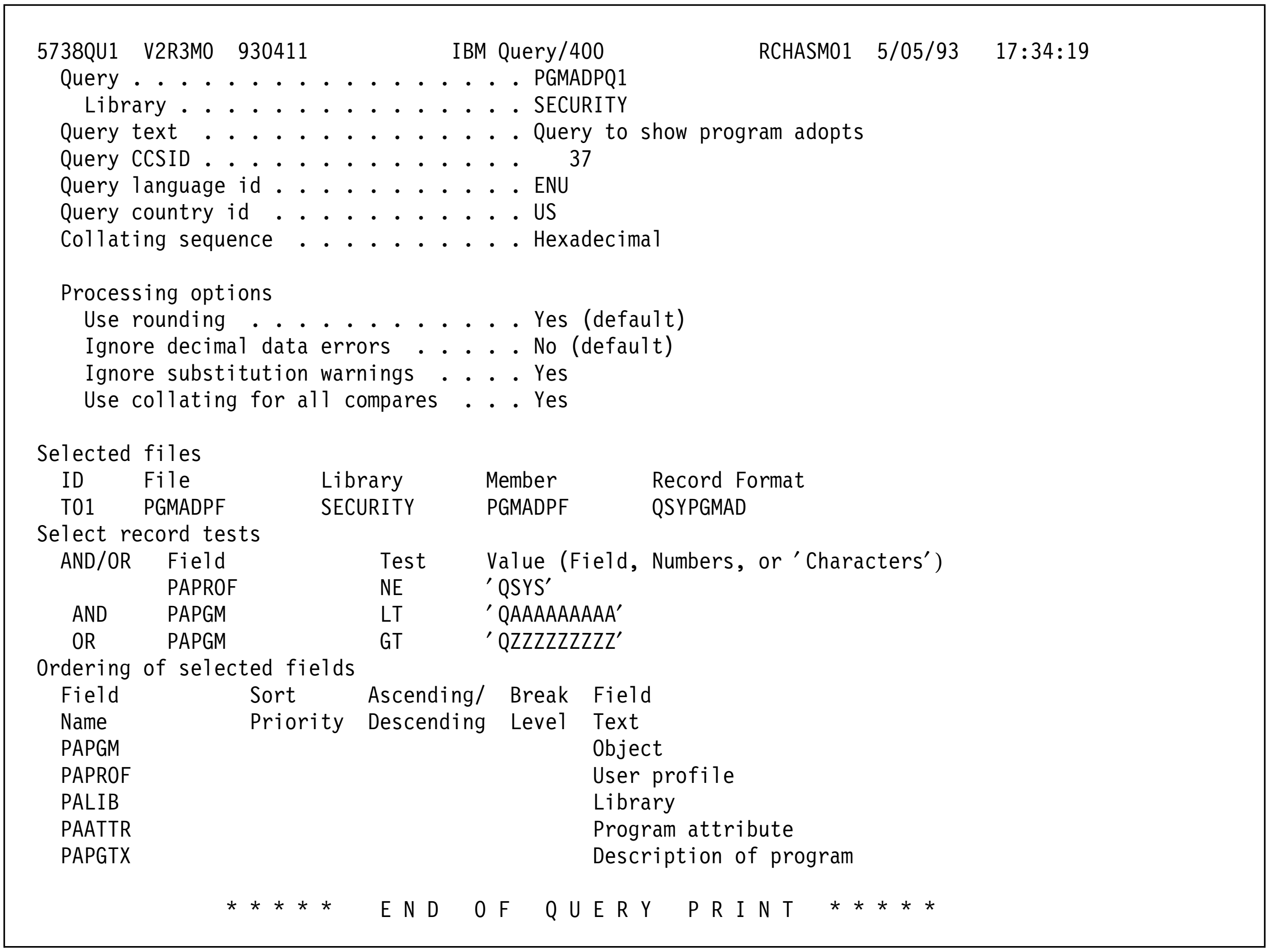
Figure 14-12. Report on All Authorization Lists

Adopted Authorities: The users can run a program if they have the necessary authority to do so. If a program has been created with USER(*OWNER) USEADPAUT(*YES), all authorities of the owner of the program are adopted. These authorities are in addition to the authorities of the user. Programs that execute with adopted authority should, therefore, be monitored.

The information on adopted authority is obtained by using the Display Program Adopt (DSPPGMADP) command. This command allows the input of one user profile at a time. The control language program in Figure 14-13 on page 14-30 places all the user profiles on the system in an outfile, in this case USRPRF. This file is then used as input to the second part of the program where each user profile is read and used as the profile parameter for the DSPPGMADP command to create a second outfile, PGMADPF. A query is then run against PGMADPF to produce the report in Figure 14-15 on page 14-32 using the query definition shown in Figure 14-14 on page 14-31.

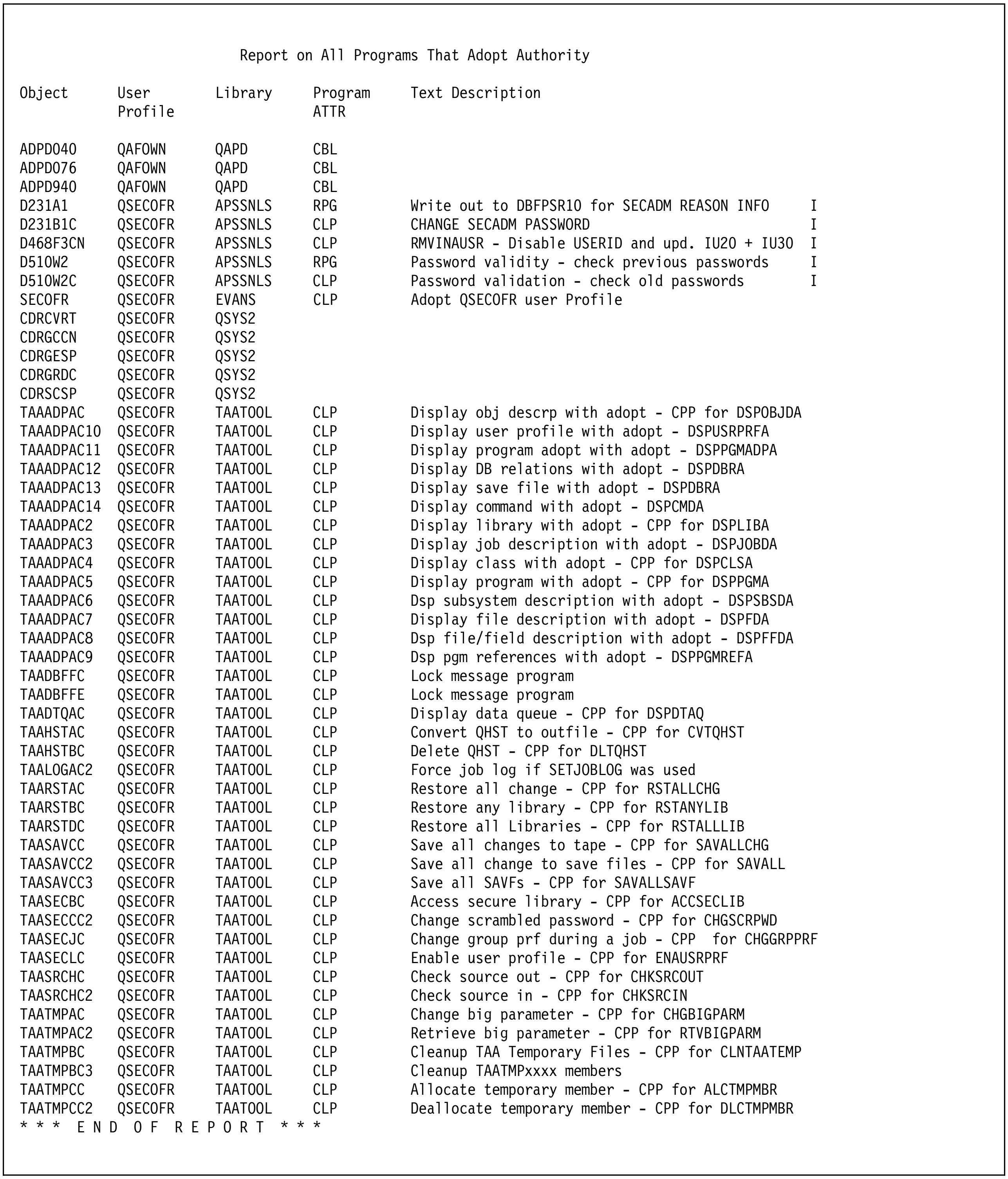
```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
  100            PGM
  200 /*****************************************************************************/
  300 /*  PGMADPCL1: This program shows all programs that adopt authorities        */
  400 /*           and the owners of these programs. The DSPPGMADP command        */
  500 /*           only allows the input of one User Profile at a time.           */
  600 /*           This program uses the DSPUSRPRF command to output all          */
  700 /*           user profiles to an OUTFILE. The user profiles are then        */
  800 /*           read sequentially into the DSPPGMADP command and is            */
  900 /*           output to an OUTFILE with the *ADD option.  A query can        */
 1000 /*           then be run against this file to produce a report.             */
 1200 /*                                                                          */
 1300 /*****************************************************************************/
 1400            DCLF       FILE(QADSPUPB) /* Format of the Display User +
 1500                                        Profile database file */
 1600            DSPUSRPRF  USRPRF(*ALL) TYPE(*BASIC) OUTPUT(*OUTFILE) +
 1700                         OUTFILE(SECURITY/USRPRF) /* Display *ALL +
 1800                         User Profile */
 1900            OVRDBF     FILE(QADSPUPB) TOFILE(SECURITY/USRPRF) /* +
 2000                         Override System supplied format file to +
 2100                         new file created */
 2200            DLTF       FILE(SECURITY/OBJAUTF) /* Delete file if it +
 2300                         already exists */
 2400            MONMSG     MSGID(CPF2105) /* Ignore if not found */
 2500  READ:     RCVF       /* Read each record on the file created */
 2600            MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
 2700            DSPPGMADP  USRPRF(&UPUPRF) OUTPUT(*OUTFILE) +
 2701                         OUTFILE(SECURITY/PGMADPF) OUTMBR(*FIRST *ADD) +
 2702                         /* Display Program adopts for each user +
 2703                         profile */
 3200            GOTO       CMDLBL(READ)
 3300  EOF:      ENDPGM
                          * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 14-13. Control Language Program to Display Program Adopts



```
5738QU1  V2R3M0  930411            IBM Query/400            RCHASM01  5/05/93   17:34:19
  Query . . . . . . . . . . . . . . . . . . PGMADPQ1
    Library . . . . . . . . . . . . . . . . SECURITY
  Query text  . . . . . . . . . . . . . . . Query to show program adopts
  Query CCSID . . . . . . . . . . . . . . .    37
  Query language id . . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . . US
  Collating sequence  . . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . . No (default)
    Ignore substitution warnings  . . . . . Yes
    Use collating for all compares  . . . . Yes

Selected files
  ID     File           Library       Member        Record Format
  T01    PGMADPF        SECURITY      PGMADPF       QSYPGMAD
Select record tests
  AND/OR   Field             Test     Value (Field, Numbers, or 'Characters')
           PAPROF            NE       'QSYS'
   AND     PAPGM             LT       'QAAAAAAAAA'
   OR      PAPGM             GT       'QZZZZZZZZZ'
Ordering of selected fields
  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text
  PAPGM                                       Object
  PAPROF                                      User profile
  PALIB                                       Library
  PAATTR                                      Program attribute
  PAPGTX                                      Description of program

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 14-14. Query Definition Showing Program Adopts



```
                           Report on All Programs That Adopt Authority

Object      User        Library     Program     Text Description
            Profile                 ATTR

ADPD040     QAFOWN      QAPD        CBL
ADPD076     QAFOWN      QAPD        CBL
ADPD940     QAFOWN      QAPD        CBL
D231A1      QSECOFR     APSSNLS     RPG         Write out to DBFPSR10 for SECADM REASON INFO     I
D231B1C     QSECOFR     APSSNLS     CLP         CHANGE SECADM PASSWORD                           I
D468F3CN    QSECOFR     APSSNLS     CLP         RMVINAUSR - Disable USERID and upd. IU20 + IU30  I
D510W2      QSECOFR     APSSNLS     RPG         Password validity - check previous passwords     I
D510W2C     QSECOFR     APSSNLS     CLP         Password validation - check old passwords        I
SECOFR      QSECOFR     EVANS       CLP         Adopt QSECOFR user Profile
CDRCVRT     QSECOFR     QSYS2
CDRGCCN     QSECOFR     QSYS2
CDRGESP     QSECOFR     QSYS2
CDRGRDC     QSECOFR     QSYS2
CDRSCSP     QSECOFR     QSYS2
TAAADPAC    QSECOFR     TAATOOL     CLP         Display obj descrp with adopt - CPP for DSPOBJDA
TAAADPAC10  QSECOFR     TAATOOL     CLP         Display user profile with adopt - DSPUSRPRFA
TAAADPAC11  QSECOFR     TAATOOL     CLP         Display program adopt with adopt - DSPPGMADPA
TAAADPAC12  QSECOFR     TAATOOL     CLP         Display DB relations with adopt - DSPDBRA
TAAADPAC13  QSECOFR     TAATOOL     CLP         Display save file with adopt - DSPDBRA
TAAADPAC14  QSECOFR     TAATOOL     CLP         Display command with adopt - DSPCMDA
TAAADPAC2   QSECOFR     TAATOOL     CLP         Display library with adopt - CPP for DSPLIBA
TAAADPAC3   QSECOFR     TAATOOL     CLP         Display job description with adopt - DSPJOBDA
TAAADPAC4   QSECOFR     TAATOOL     CLP         Display class with adopt - CPP for DSPCLSA
TAAADPAC5   QSECOFR     TAATOOL     CLP         Display program with adopt - CPP for DSPPGMA
TAAADPAC6   QSECOFR     TAATOOL     CLP         Dsp subsystem description with adopt - DSPSBSDA
TAAADPAC7   QSECOFR     TAATOOL     CLP         Display file description with adopt - DSPFDA
TAAADPAC8   QSECOFR     TAATOOL     CLP         Dsp file/field description with adopt - DSPFFDA
TAAADPAC9   QSECOFR     TAATOOL     CLP         Dsp pgm references with adopt - DSPPGMREFA
TAADBFFC    QSECOFR     TAATOOL     CLP         Lock message program
TAADBFFE    QSECOFR     TAATOOL     CLP         Lock message program
TAADTQAC    QSECOFR     TAATOOL     CLP         Display data queue - CPP for DSPDTAQ
TAAHSTAC    QSECOFR     TAATOOL     CLP         Convert QHST to outfile - CPP for CVTQHST
TAAHSTBC    QSECOFR     TAATOOL     CLP         Delete QHST - CPP for DLTQHST
TAALOGAC2   QSECOFR     TAATOOL     CLP         Force job log if SETJOBLOG was used
TAARSTAC    QSECOFR     TAATOOL     CLP         Restore all change - CPP for RSTALLCHG
TAARSTBC    QSECOFR     TAATOOL     CLP         Restore any library - CPP for RSTANYLIB
TAARSTDC    QSECOFR     TAATOOL     CLP         Restore all Libraries - CPP for RSTALLLIB
TAASAVCC    QSECOFR     TAATOOL     CLP         Save all changes to tape - CPP for SAVALLCHG
TAASAVCC2   QSECOFR     TAATOOL     CLP         Save all change to save files - CPP for SAVALL
TAASAVCC3   QSECOFR     TAATOOL     CLP         Save all SAVFs - CPP for SAVALLSAVF
TAASECBC    QSECOFR     TAATOOL     CLP         Access secure library - CPP for ACCSECLIB
TAASECCC2   QSECOFR     TAATOOL     CLP         Change scrambled password - CPP for CHGSCRPWD
TAASECJC    QSECOFR     TAATOOL     CLP         Change group prf during a job - CPP  for CHGGRPPRF
TAASECLC    QSECOFR     TAATOOL     CLP         Enable user profile - CPP for ENAUSRPRF
TAASRCHC    QSECOFR     TAATOOL     CLP         Check source out - CPP for CHKSRCOUT
TAASRCHC2   QSECOFR     TAATOOL     CLP         Check source in - CPP for CHKSRCIN
TAATMPAC    QSECOFR     TAATOOL     CLP         Change big parameter - CPP for CHGBIGPARM
TAATMPAC2   QSECOFR     TAATOOL     CLP         Retrieve big parameter - CPP for RTVBIGPARM
TAATMPBC    QSECOFR     TAATOOL     CLP         Cleanup TAA Temporary Files - CPP for CLNTAATEMP
TAATMPBC3   QSECOFR     TAATOOL     CLP         Cleanup TAATMPxxxx members
TAATMPCC    QSECOFR     TAATOOL     CLP         Allocate temporary member - CPP for ALCTMPMBR
TAATMPCC2   QSECOFR     TAATOOL     CLP         Deallocate temporary member - CPP for DLCTMPMBR
* * *  E N D  O F  R E P O R T  * * *
```

Figure 14-15. Report Showing Program Adopts

The report in Figure 14-15 gives a listing of all programs, except system programs, that run with adopted authority. The owner of each program is shown in the User Profile column. By using the Display Program References (DSPPGMREF) command for the programs listed in the report, the files used in each program can be displayed or printed.

Using this report, the program adoption can be monitored and controlled.

Auditing Successful Access to a Database File: A database file is either a physical file or a logical file, and a logical file is just simply a view of one or more physical files. Access to a database file is either to read, to write (to add or to update), or to delete. Writing and deleting cause changes to a file.

The critical nature of some database files may require that all accesses to the files are recorded, regardless of whether the attempts are successful or not. Unsuccessful attempts to access a database file are violations and are recorded in the history log or the audit journal. Monitoring successful access, however, requires a different approach.

Successful access that changes a physical file can be recorded by having that particular file audited. If a logical file is dependent on several physical files, it is recommended that all the physical files be audited.

You can utilize OS/400 object auditing function to audit every access of a specified database file, see Section 6.5.2, “Object Auditing” on page 6-19 for more information.

You can create your own journal or use system-provided journal QAUDJRN to identify the journal receiver where the event is recorded. The Display Journal (DSPJRN) command allows you to convert the journal entries (contained in one or more receivers) into a suitable form for output. Output can be displayed on the screen, printed, or written to an outfile.

To get a listing of all journals on the system, enter the following:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYP(*JRN) OUTPUT(*PRINT)
```

If you are journaling and want to print all information about a particular file, enter the following:

```
DSPJRN JRN(library/journal) FILE((library/file)) OUTPUT(*PRINT)
```

If journal JOURNAL in library SECURITY is used to record information about file USRINLC (also in library SECURITY), the command would be:

```
DSPJRN JRN(SECURITY/JOURNAL) FILE((SECURITY/USRINLC)) OUTPUT(*PRINT)
```

If your output is written to an outfile, you can then analyze the audit data in the outfile by using Query/400, a user written program, or the DSPAUDLOG command. The DSPAUDLOG command source is provided in library QUSRTOOL. Section 6.4.6, “Display Audit Log Command” on page 6-12 has a more detailed information about how to create and use DSPAUDLOG command.

Auditing Security Configuration Changes: The outfiles produced by security related commands for audit purposes should be saved for later comparison. Queries can be run against the current and previous outfiles to check for the changes, for example, in user profiles or authorization lists.

14.7 Securing an AS/400

Securing an AS/400 can be divided into three steps:

1. Find the security status of the system and the company
2. Make a plan of what must be done, and implement it
3. Audit what is happening on the system

It is basically the same procedures you will go through if you inherited a house in a remote place you have never been. Let us look at the similarities:

14.7.1 Find the Status

The Neighborhood: You take a trip to inspect the house. The first thing you will notice is the neighborhood, and you must learn more about it. As an auditor you will ask questions about routines and policies in the company you are going to work with. These routines and policies are essential and must be verified.

The Foundation: The entire house rests on it, so it better be in good shape. You should always start with the system values.

The Roof: If the roof leaks there is no reason to redecorate or fix anything. The user profiles represent the roof, and it usually leaks. The choice of user classes and use of special authorities must be verified.

Windows and Doors: Windows and doors give free access to the house when not properly closed and secured. The entrance door must have some kind of lock. As an auditor you must verify the access to the applications.

Ownership: When the doors and windows are taken care of there is a question about who should have the keys to the different doors in the house. The owner is the one to pass out keys (authorization) to the doors he/she owns. Ownership is vital in an AS/400. The owner is almighty and cannot be restricted from any action on owned objects.

Fence and Gate: The fence and gate can give a ″controlled″ access to your property and the house. As an auditor you must verify the use of communications lines.

Tenants: If you have tenants in the house you probably will not let them have access to your private part of the house, and all your belongings. There will be some restrictions set up by yourself. As an auditor you must find out how programmers work on the AS/400. The programmers may be employed by the company, or be business partners. Many places you will find that they can do whatever they want, and nobody seems to care. They may all be honest and reliable people, but that is beside the point for an auditor.

14.7.2 Implementing Better Security

When checking your house you make notes about everything you find so you can estimate what must be done to bring the house up to an acceptable level. When you check out your house you will make notes of what is good, and what needs attention. As an auditor you will do the same, and these notes are the base for the status report. In this report you point out not only what you have found and what must be done to improve the situation, but also the consequences. Just as with your house you must try to estimate the cost and the benefits.

If you are the one to implement the changes; make it into a project with participants, responsibilities, time frames. Include everything.

14.7.3 Daily Auditing

When you have invested in improving the standard of your house you will not let it fall into decay. Since you live somewhere far away you cannot keep an eye on it yourself, so you pay a person to look after it and report to you. The audit journal must be started, and you must ensure that the right persons get the reports that concern them, and that they are able to understand the reports and know when to take action.

14.8 The Auditors Menu

When auditing an AS/400 there is always a question of how much time you can put into it. There is no standard answer to that. Every case must be evaluated separately, because every company is different.

At the present time the following is available to help you to audit an AS/400:

- Security/400
- Programs in QUSRTOOL. There are quite a few there that can be used
- Programs described in this book, and in referenced manuals

You should make you own ″auditing package″. We also recommend that you create a shell-document for your status report. You must have one, and it will help you analyze the job before you start.

Typically, every full system audit covers each of the following items. The time available will determine the amount of detail you go into in each area, and the depth of recommendations you make.

In the following we have listed some options under different menus. They do in no way represent a complete list, but may give you an idea of what to implement in your ″auditing package″.

The Foundation Menu

1. Security Goals and Objectives

   There is no point in attempting to audit an AS/400 system if you have nothing to compare it against. Every company should have a security plan, and this will define the scope of your audit. If there is no security plan, it is important to define one first **before you begin the audit**.

2. System values related to security

The report in Figure 14-2 on page 14-15 or Figure 15-7 on page 15-15 tells a lot, such as:

- If QAUDJRN is used
- If an inactive terminal is logged off
- If multiple sessions are allowed
- If users with *SECOFR or *SERVICE special authorities can log on from all terminals
- How invalid signon attempts are handled
- How the password may be built
- How DSPT is handled
- The security level, and what to expect from user profiles

The Roof Menu

1. Registered user profiles

   Sorted on the last date the password was changed, such a report tells if the information flows as it should within the company. If it does, the report tells if rules are followed. DSPSECRVW in QUSRTOOL has many different reports on user profiles.

2. Users with special authorities.

   A report like the one in Figure 14-9 on page 14-25 or Figure 15-9 on page 15-18 may be used.

3. Group profiles

   A report like the one in Figure 14-4 on page 14-19 may be used.

4. Group profiles and ownership.

   If the group profile owns objects, all the members of the group have too high authorization to those objects.

5. Authority to the user profiles

   A report like the one in Figure 15-4 on page 15-11 may be used.

6. Adopted authority

   A report like the one in Figure 14-15 on page 14-32 or Figure 15-6 on page 15-14 may be used.

The Windows and Doors Menu

1. Access to the libraries

   List public authority and ownership. A report like the one in Figure 15-1 on page 15-7 or Figure 15-2 on page 15-8 may be used.

2. Objects secured by authorization lists

   A report like the one in Figure 15-8 on page 15-16 may be used.

3. Object owners within a library

   List the objects within a library that are not owned by the library owner.

4. USER parameter in the job descriptions.

The Fence and Gate Menu

1. Network attributes
2. List of communication lines
3. List of remote controllers
4. List of APPC devices
5. Print of the subsystems
6. Local and remote configuration lists
7. Network job entries
8. System directory
9. DDM files
10. ICF files

Some companies may want to audit the OfficeVision environment, or the access to folders and their objects. This may lead to a restructuring of the folders, and a change in the backup routines, the restore strategy, and the ″Plan to protect the business processes″.

You may want to have an Office Menu to cover this.

Chapter 15. IBM Security/400

A package of tools is now available to aid in your AS/400 security administration! IBM Security/400 (5764-006) is licensed on an ″as is″ basis at a very reasonable cost. For ordering information, contact your local branch office.

IBM Security/400 allows the system administrator to monitor and track system users to ensure they are in compliance with their security policy. It will accomplish this task by providing the ability to create reports on files, libraries, command, user profiles and job descriptions. In addition, it provides reporting on programs that adopt authority, user profiles with access authority, authorization lists, and network values. This reporting will help control system compliance and also identify areas that may require tighter controls.

15.1.1 IBM Security/400 History

IBM Rochester, home of the AS/400, uses the AS/400 extensively throughout its production and manufacturing areas. An entire I/S area is dedicated to supporting the AS/400 in its production environment. Over time, IBM Rochester′s I/S shop has identified security areas that require additional reports and examination that are not provided by the AS/400 operating system. The I/S shop developed a number of applications to fulfill this need. IBM would like share the knowledge and expertise gained in its own I/S environment and decided to offer you the code. The group of applications is being packaged as IBM Security/400.

IBM Security/400 allows you to tailor the applications to your individual environment by supplying the source code at no additional cost. This allows you the flexibility to design and implement your own security control for your environment. To aid in tailoring your applications, IBM Security/400 includes:

- Installation Instructions
- Technical Documentation
- Object Code
- Source Code

15.1.2 IBM Security/400 Components

The major components of IBM Security/400 are:

Auditor

The Auditor function is menu driven. It provides the following functions:

- Reports public authority to libraries, files, commands, job descriptions, and user profiles
- Reports on specific system values and network attributes
- Reports contents of authorities on authorization lists
- Reports programs that adopt authority
- Reports all user profiles on the system (optionally reported by profile attributes)
- Reports all the attributes contained within QAUDJRN (change to profiles, invalid signon attempts, and restore objects report)

Remove Inactive User

Users who have not changed their password in the last 186 days (6 months) will have their user profiles disabled. Two notifications are sent to the security administrator as a warning that the user profile has been disabled and will be deleted from the system if no action is taken.

Use SECADM Profile

This utility is designed to give a user the ability to sign on to the system with a profile which has *SECADM authority. The user profile is named SECADM to avoid any confusion. All activity is tracked while the user is signed on with this profile. This allows queries to be run against usage of the profile. After the user signs off the system, the SECADM user profile′s password is changed to *NONE. If the user wants to sign on to the system again using the SECADM user profile, the SECADM command must be executed again to reset the password.

Universal Access *NONE

This program revokes *PUBLIC authority from root folders and libraries, with the exception of registered objects. Root level folders are the very top level folders. For instance, if you have a folder VANRIPER, which is not contained within another folder, and contains a folder SUSAN, then VANRIPER is considered the root level folder and SUSAN is the subfolder. Objects may be registered by having the object name entered into a database file. If the object is not registered and has *PUBLIC authority, a warning notification is sent to the owner. The owner must ensure that proper authority is granted to all users who need access to the object. If the owner does not justify the need for public authority to this object within 10 days, the public authority will be changed to *EXCLUDE.

Note

Securing root level folders from public access does not necessarily secure all folders contained within the folder. Refer to Chapter 13, “OfficeVision/400” on page 13-1 for more information.

Systematic Logon Reporting

This utility is designed to report excessive invalid signon attempts and user profile disables. (Excessive is determined by the user at installation time.) The utility uses the QAUDJRN journal to extract the information needed to count the number of invalid signon attempts and revokes (user profile disables caused by exceeding the limit of invalid signon attempts). If the invalid signon attempt exceeds the value stored in the data access, a message is distributed to the system administrator.

15.1.3 Installation Instructions

IBM Security/400 is packaged and installed in the Licensed Program Product format. When you receive your media from IBM, perform the following functions:

1.____ Mount the media on your AS/400

2.____ Sign on to your system with QSECOFR user profile

3.____ Enter the command:

```
RSTLICPGM LICPGM(2R69801) DEV(TAPxx)
```

4. ____ Check message queue QSYSOPR for any pertinent messages

All objects and source code is contained in two libraries:

- APSSNLS
- AUDITOR

The Auditor portion of IBM Security/400 is ready to use. The remainder of the applications require further installation. If you wish to utilize these additional applications, refer to the IBM Security/400 Installation Instructions, which are provided with the product, for further installation details.

Note

The installation process may change current auditing system values. Any changes to system values will be logged in the QSYSOPR message queue.

The installation process will create two user profiles on your system which have special authorities. They are:

- AUDITOR - owns all objects in the product
- AUDSECOFR - user profile under which jobs run

15.2 AUDITOR

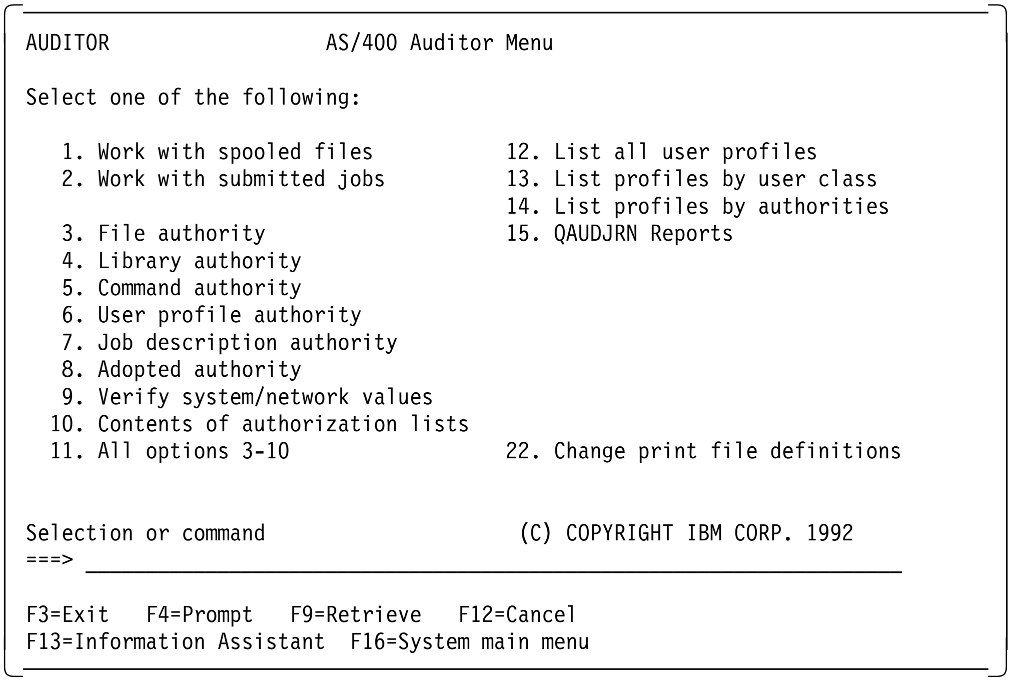
AUDITOR is a menu-driven application which allows an auditor to select menu items to generate a series of reports.

15.2.1 AUDITOR Detailed Description

To access the Auditor menu, sign on as QSECOFR (or equivalent) and execute the following commands:

1.____ ADDLIBLE AUDITOR

2.____ AUDITOR



```
AUDITOR                  AS/400 Auditor Menu

Select one of the following:

   1. Work with spooled files           12. List all user profiles
   2. Work with submitted jobs          13. List profiles by user class
                                        14. List profiles by authorities
   3. File authority                    15. QAUDJRN Reports
   4. Library authority
   5. Command authority
   6. User profile authority
   7. Job description authority
   8. Adopted authority
   9. Verify system/network values
  10. Contents of authorization lists
  11. All options 3-10                  22. Change print file definitions


Selection or command                     (C) COPYRIGHT IBM CORP. 1992
===> _____________________________________________________________________

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=Information Assistant  F16=System main menu
```

AUDITOR menu options include:

- 1. Work with Spooled files - allows you to work with the reports that were generated by AUDITOR.
- 2. Work with Submitted Jobs - allows you to work with the jobs that were submitted by AUDITOR.
- 3. File Authority - submits a job to batch which will generate a report of *PUBLIC authority to ALL files on the system.
- 4. Library Authority - submits a job to batch which will generate a report of *PUBLIC authority to ALL libraries on the system.
- 5. Command Authority - submits a job to batch which will generate a report of *PUBLIC authority to ALL commands on the system.
- 6. User Profile Authority - submits a job to batch which will generate a report of *PUBLIC authority to ALL user profiles on the system.
- 7. Job Description Authority - submits a job to batch which generates a report of *PUBLIC access to job descriptions. This option differs from previous reports because it further subsets the report and excludes any job descriptions in libraries in which the *PUBLIC authority to the library is *EXCLUDE. In addition, the report only contains job descriptions that have the USER parameter set to a value other than *RQD.
- 8.Adopted authority - submits a job to batch which will generate a report that lists programs that adopt authority.
- 9. Verify System Values/Network Values - submits a job to batch which will generate a list of security related system values and network attributes.
- 10. Contents of Authorization Lists - submits a job to batch which will generate a report of ALL authorization lists.
- 11. All options 3-10 - submits a job, which in turn submits jobs for options 3-10 to batch.

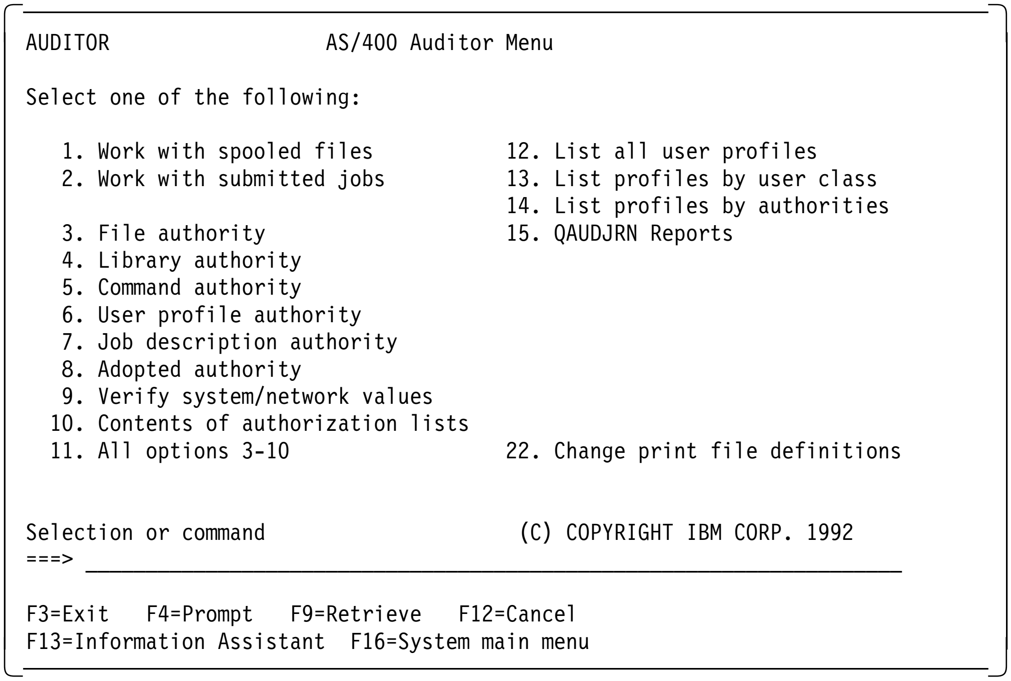
- 12. List All User Profiles - submits a job to batch which generates a report of all BASIC information for ALL user profiles on the system.
- 13. List User Profiles by Class - presents a screen to you to select which class of user you want to choose. After you select the user profiles you want to be contained in the report, a job is submitted to batch which generates a report of all BASIC information for the classes of user profiles selected. You may select more than one class to appear on the report.
- 14. List user profiles by Authority - presents a screen to you to select which user′s with the particular special authorities you want to appear in the report. For instance, you may want to generate a report of only those users on the system which have *ALLOBJ special authority.
- 15. QAUDJRN Reports - presents a screen to you which allows you to select reports to be run over the system audit journal, QAUDJRN. You may select a date range to report on. You may also request to have the journal receiver be changed at this time. You may specify a journal receive name, or let the system generate a new journal receive name for you.
- 22. Change Print File Definitions - allows you to tailor all AUDITOR print files to your print environment.

15.2.2 AUDITOR - Interactive Environment

To access the AUDITOR portion of the package interactively, execute the following commands:

```
AUDITOR/AUDITOR
```

You will be presented with the following menu:



```
AUDITOR                  AS/400 Auditor Menu

Select one of the following:

   1. Work with spooled files          12. List all user profiles
   2. Work with submitted jobs         13. List profiles by user class
                                       14. List profiles by authorities
   3. File authority                   15. QAUDJRN Reports
   4. Library authority
   5. Command authority
   6. User profile authority
   7. Job description authority
   8. Adopted authority
   9. Verify system/network values
  10. Contents of authorization lists
  11. All options 3-10                 22. Change print file definitions


Selection or command                    (C) COPYRIGHT IBM CORP. 1992
===> ____________________________________________________________________

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=Information Assistant  F16=System main menu
```

You may choose any of the individual options, or group a subset of options. A subset of options 3-10 is provided using option 11. All reports submitted via the AUDITOR menu will be submitted to batch using the QGPL/QBATCH job queue.

15.2.3 AUDITOR - Batch Environment

Many of the options listed on the AUDITOR menu can be submitted directly to batch via the job scheduler. For instance, if you want to run the QAUDRPT report on a monthly basis, you can use the following command to generate the report automatically:

```
ADDJOBSCDE JOB(QAUDRPT) CMD(QAUDRPT BEGDATE('MM/DD/YY')
  ENDDATE('MM/DD/YY') AUTOMATE(Y) CHGJRNRCV(Y)) FRQ(*MONTHLY)
  SCDDATE(*MONTHEND) SCDTIME(0500) JOBD(AUDITOR/AUDSECOFR)
```

Note

Some of the options require parameters to be passed. Please refer to the documentation supplied with the IBM Security/400 more details.

15.2.4 AUDITOR - Locating and Printing Reports

IBM Security/400 is shipped with a default output queue to store all reports generated. The output queue is AUDITOR/AUDITOR. You can use option 1 from the auditor menu to work with the output queue. You may also use the command:

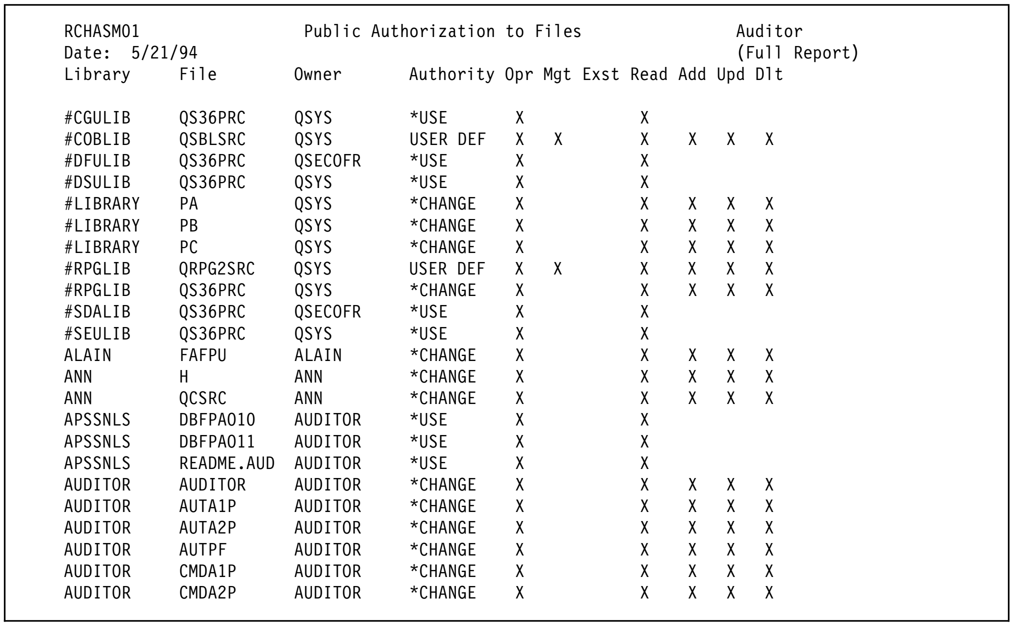
```
WRKOUTQ AUDITOR/AUDITOR
```

to work with the output queue.

15.2.5 Auditor - File Authority Report (FILA1P, FILA2P)

The File Authority Report lists the public authority all files on the system. It generates two reports:

- FILA1P - Lists public authority to all files on the system
- FILA2P - Lists public authority to all files on the system that have changed since the last time this option was run



```
RCHASM01                 Public Authorization to Files                Auditor
Date:  5/21/94                                                        (Full Report)
Library     File        Owner       Authority Opr Mgt Exst Read Add Upd Dlt

#CGULIB     QS36PRC     QSYS        *USE       X            X
#COBLIB     QSBLSRC     QSYS        USER DEF   X   X        X    X   X   X
#DFULIB     QS36PRC     QSECOFR     *USE       X            X
#DSULIB     QS36PRC     QSYS        *USE       X            X
#LIBRARY    PA          QSYS        *CHANGE    X            X    X   X   X
#LIBRARY    PB          QSYS        *CHANGE    X            X    X   X   X
#LIBRARY    PC          QSYS        *CHANGE    X            X    X   X   X
#RPGLIB     QRPG2SRC    QSYS        USER DEF   X   X        X    X   X   X
#RPGLIB     QS36PRC     QSYS        *CHANGE    X            X    X   X   X
#SDALIB     QS36PRC     QSECOFR     *USE       X            X
#SEULIB     QS36PRC     QSYS        *USE       X            X
ALAIN       FAFPU       ALAIN       *CHANGE    X            X    X   X   X
ANN         H           ANN         *CHANGE    X            X    X   X   X
ANN         QCSRC       ANN         *CHANGE    X            X    X   X   X
APSSNLS     DBFPA010    AUDITOR     *USE       X            X
APSSNLS     DBFPA011    AUDITOR     *USE       X            X
APSSNLS     README.AUD  AUDITOR     *USE       X            X
AUDITOR     AUDITOR     AUDITOR     *CHANGE    X            X    X   X   X
AUDITOR     AUTA1P      AUDITOR     *CHANGE    X            X    X   X   X
AUDITOR     AUTA2P      AUDITOR     *CHANGE    X            X    X   X   X
AUDITOR     AUTPF       AUDITOR     *CHANGE    X            X    X   X   X
AUDITOR     CMDA1P      AUDITOR     *CHANGE    X            X    X   X   X
AUDITOR     CMDA2P      AUDITOR     *CHANGE    X            X    X   X   X
```

Figure 15-1. Sample FILA1P Report

Table 15-1. File Authority Report - Generated 02/01/94

Object	Library	*PUBLIC Authority to Object	*PUBLIC Authority to Library	Last Date Report Run	*PUBLIC Authority Last Change Date	Appear in FILA1P	Appear in FILA2P
FILE1	LIB1	*EXCLUDE	*EXCLUDE	01/01/94	12/15/93	Y	N
FILE2	LIB1	*EXCLUDE	*EXCLUDE	01/01/94	01/15/94	Y	Y
FILE3	LIB1	*CHANGE	*EXCLUDE	01/01/94	12/15/93	Y	N
FILE4	LIB2	*CHANGE	*ALL	01/01/94	01/15/94	Y	Y

Example: File LIBX/FILEY exists on your system and its public authority is *PUBLIC *EXCLUDE (refer to Table 15-1). The public authority has not changed since that last time you ran this report. If you run the File Authority Report on February 1, 1994 it will list the public authority to file LIBX/FILEY as *EXCLUDE in the FILA1P report. It will not list the file in the FILA2P report, because you haven′t changed the authority since the last time you ran this report.

The public authority to file LIBX/FILEY is changed on January 15, 1994 to *CHANGE. On February 1, 1994, you run the File Authority Report again. This time, the file LIBX/FILEY will appear on the FILA1P report as *PUBLIC *CHANGE. It will also appear on the FILA2P report as *PUBLIC *CHANGE.

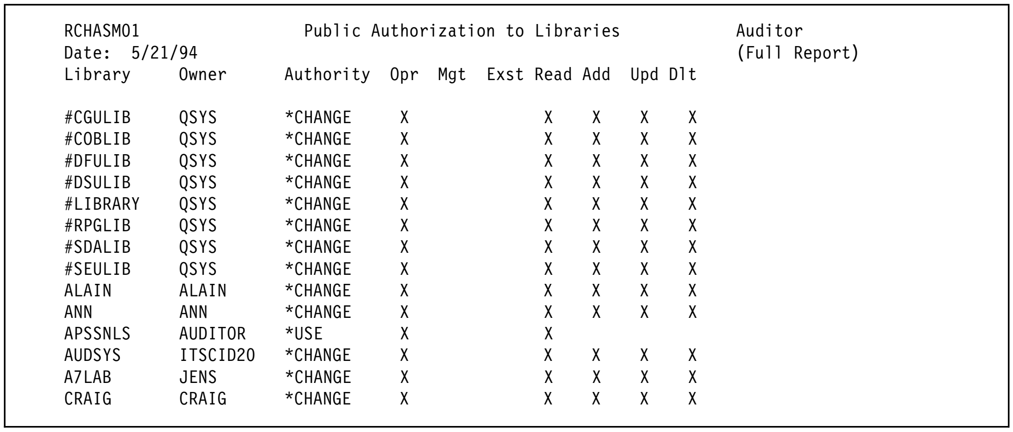
Suggestion - Use the FILA2P report on a regular basis to review changes to file authority on your report. Unless you have numerous file authority changes, this report should be short and easy to read.

Use the FILA1P report to perform in depth reviews of your system′s file authority. The report can become very large and reviewing can be tedious work.

15.2.6 Auditor - Library Authority Report (LIBA1P, LIBA2P)

The Library Authority Report lists the public authority to all libraries on the system. It generates two reports:

- LIBA1P - Lists public authority to all libraries on the system
- LIBA2P - Lists public authority to all libraries on the system that have changed since the last time this option was run



```
RCHASM01                 Public Authorization to Libraries            Auditor
Date:  5/21/94                                                        (Full Report)
Library     Owner      Authority  Opr  Mgt  Exst Read Add  Upd Dlt

#CGULIB     QSYS       *CHANGE     X              X    X    X    X
#COBLIB     QSYS       *CHANGE     X              X    X    X    X
#DFULIB     QSYS       *CHANGE     X              X    X    X    X
#DSULIB     QSYS       *CHANGE     X              X    X    X    X
#LIBRARY    QSYS       *CHANGE     X              X    X    X    X
#RPGLIB     QSYS       *CHANGE     X              X    X    X    X
#SDALIB     QSYS       *CHANGE     X              X    X    X    X
#SEULIB     QSYS       *CHANGE     X              X    X    X    X
ALAIN       ALAIN      *CHANGE     X              X    X    X    X
ANN         ANN        *CHANGE     X              X    X    X    X
APSSNLS     AUDITOR    *USE        X              X
AUDSYS      ITSCID20   *CHANGE     X              X    X    X    X
A7LAB       JENS       *CHANGE     X              X    X    X    X
CRAIG       CRAIG      *CHANGE     X              X    X    X    X
```

Figure 15-2. Sample LIBA1P Report

Table 15-2. Library Report - Report Generated on 02/01/94

Library	*PUBLIC Authority to Library	Last Date Report Run	*PUBLIC Authority Last Change Date	Appear in LIBA1P	Appear in LIBA2P
LIB1	*EXCLUDE	01/01/94	12/15/93	Y	N
LIB2	*EXCLUDE	01/01/94	01/15/94	Y	Y
LIB3	*CHANGE	01/01/94	12/15/93	Y	N
LIB4	*CHANGE	01/01/94	01/15/94	Y	Y

Example: Library LIBX exists on your system and its public authority is *EXCLUDE. The public authority has not changed since that last time you ran this report. If you run the Library Authority Report on February 1, 1994 it will list the public authority to LIBX as *EXCLUDE in the LIBA1P report. It will not list the library in the LIBA2P report, because you haven′t changed the authority since the last time you ran this report.

The public authority to library LIBX is changed on January 15, 1994 to *CHANGE. On February 1, 1994, you run the Library Authority Report again. This time, the library LIB1 will appear on the LIBA1P report as *PUBLIC *CHANGE. It will also appear on the LIBA2P report as *PUBLIC *CHANGE.

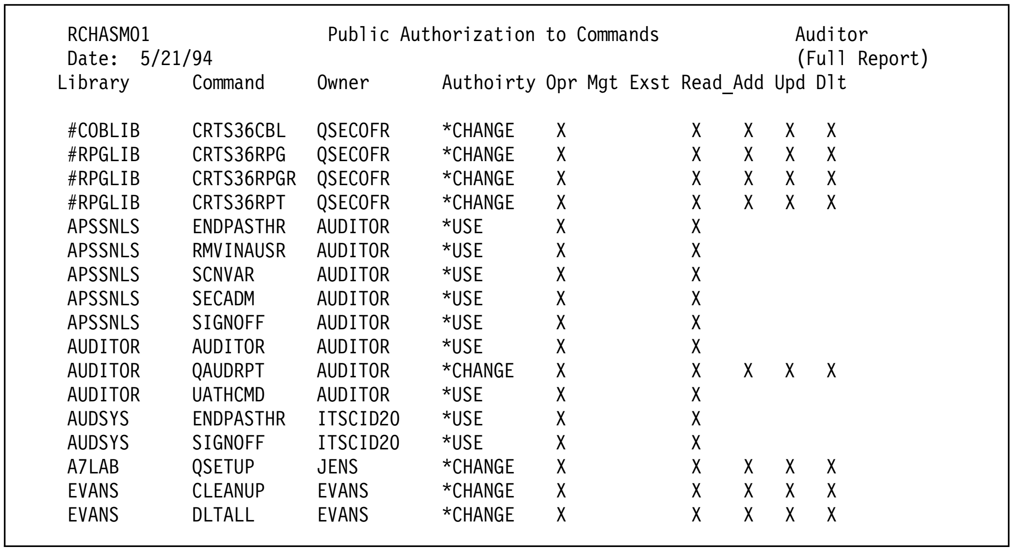
Suggestion - Use the LIBA2P report on a regular basis to review changes to library authority on your report. Unless you have numerous library authority changes, this report should be short and easy to read.

Use the LIBA1P report regularly to perform reviews of your system′s library authority. This report can provide very useful general information on your security posture.

15.2.7 Auditor - Command Authority Report (CMDA1P, CMDA2P)

The Command Authority Report lists the public authority to all commands on the system. It generates two reports:

- CMDA1P - Lists public authority to all commands on the system
- CMDA2P - Lists public authority to all commands on the system that have changed since the last time this option was run



```
 RCHASM01                 Public Authorization to Commands             Auditor
 Date:  5/21/94                                                        (Full Report)
Library      Command     Owner       Authoirty Opr Mgt Exst Read_Add Upd Dlt

 #COBLIB     CRTS36CBL   QSECOFR     *CHANGE    X            X    X   X   X
 #RPGLIB     CRTS36RPG   QSECOFR     *CHANGE    X            X    X   X   X
 #RPGLIB     CRTS36RPGR  QSECOFR     *CHANGE    X            X    X   X   X
 #RPGLIB     CRTS36RPT   QSECOFR     *CHANGE    X            X    X   X   X
 APSSNLS     ENDPASTHR   AUDITOR     *USE       X            X
 APSSNLS     RMVINAUSR   AUDITOR     *USE       X            X
 APSSNLS     SCNVAR      AUDITOR     *USE       X            X
 APSSNLS     SECADM      AUDITOR     *USE       X            X
 APSSNLS     SIGNOFF     AUDITOR     *USE       X            X
 AUDITOR     AUDITOR     AUDITOR     *USE       X            X
 AUDITOR     QAUDRPT     AUDITOR     *CHANGE    X            X    X   X   X
 AUDITOR     UATHCMD     AUDITOR     *USE       X            X
 AUDSYS      ENDPASTHR   ITSCID20    *USE       X            X
 AUDSYS      SIGNOFF     ITSCID20    *USE       X            X
 A7LAB       QSETUP      JENS        *CHANGE    X            X    X   X   X
 EVANS       CLEANUP     EVANS       *CHANGE    X            X    X   X   X
 EVANS       DLTALL      EVANS       *CHANGE    X            X    X   X   X
```

Figure 15-3. Sample CMDA1P Report

Table 15-3. Command Authority Report - Report Generated on 02/01/94

Object	Library	*PUBLIC Authority to Command	*PUBLIC Authority to Library	Last Date Report Run	*PUBLIC Authority Last Change Date	Appear in CMDA1P	Appear in CMDA2P
CMD1	LIB1	*EXCLUDE	*EXCLUDE	01/01/94	12/15/93	Y	N
CMD2	LIB1	*EXCLUDE	*EXCLUDE	01/01/94	01/15/94	Y	Y
CMD3	LIB1	*CHANGE	*EXCLUDE	01/01/94	12/15/93	Y	N
CMD4	LIB2	*CHANGE	*ALL	01/01/94	01/15/94	Y	Y

Example: The command LIBX/CMDY exists on your system and its public authority is *EXCLUDE. The public authority has not changed since the last time you ran this report. If you run the Command Authority Report on February 1, 1994, it will list the public authority to file LIBX/CMDY as *EXCLUDE in the CMDA1P report. It will not list the command in the CMDA2P report, because you haven′t changed the authority since the last time you ran this report.

The public authority to command LIBX/CMDY is changed on January 15, 1994 to *CHANGE. On February 1, 1994, you run the Command Authority Report again. This time, the command LIBX/CMDY will appear on the CMDA1P report as *PUBLIC *CHANGE. It will also appear on the CMDA2P report as *PUBLIC *CHANGE.

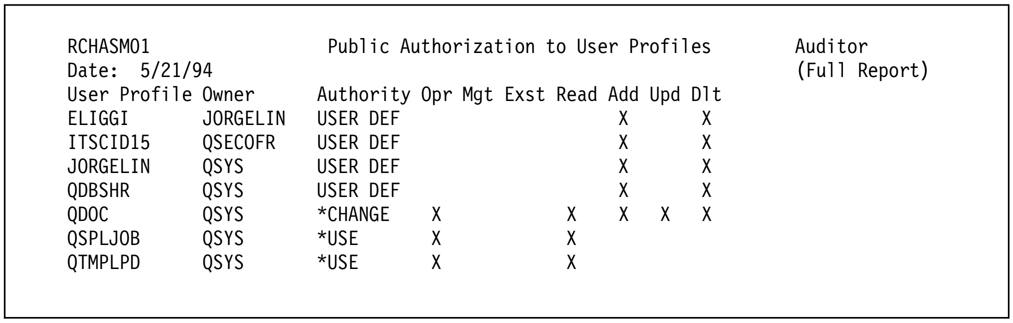
Suggestion - Use the CMDA2P report on a regular basis to review changes to command authority on your report. Unless you have numerous command authority changes, this report should be short and easy to read.

Use the CMDA1P report to perform in depth reviews of your system′s command authority. The report can become very large and reviewing can be tedious work.

15.2.8 Auditor - User Profile Authority Report (PRFA1P, PRFA2P)

The User Profile Authority Report lists the public authority user profiles on the system which do NOT have *PUBLIC *EXCLUDE. It generates two reports:

- PRFA1P - Lists public authority to all user profiles on the system
- PRFA2P - Lists public authority to all user profiles on the system that have changed since the last time this option was run



```
RCHASM01                 Public Authorization to User Profiles        Auditor
Date:  5/21/94                                                        (Full Report)
User Profile Owner      Authority Opr Mgt Exst Read Add Upd Dlt
ELIGGI       JORGELIN   USER DEF                     X       X
ITSCID15     QSECOFR    USER DEF                     X       X
JORGELIN     QSYS       USER DEF                     X       X
QDBSHR       QSYS       USER DEF                     X       X
QDOC         QSYS       *CHANGE    X            X    X   X   X
QSPLJOB      QSYS       *USE       X            X
QTMPLPD      QSYS       *USE       X            X
```

Figure 15-4. Sample PRFA1P Report

Table 15-4. User Profile Report - Report Generated on 02/01/94

Library	*PUBLIC Authority to User Profile	Last Date Report Run	*PUBLIC Authority Last Change Date	Appear in PRFA1P	Appear in PRFA2P
USRPRF1	*EXCLUDE	01/01/94	12/15/93	N	N
USRPRF2	*EXCLUDE	01/01/94	01/15/94	N	N
USRPRF3	*CHANGE	01/01/94	12/15/93	Y	N
USRPRF4	*CHANGE	01/01/94	01/15/94	Y	Y

Example 1: User profile USRPRFX exists on your system and its public authority is *CHANGE. The public authority has not changed since that last time you ran this report. If you run the User Profile Authority Report on February 1, 1994, it will list the public authority to user profile USRPRFX as *CHANGE in the PRFA1P report. It will not list the user profile in the PRFA2P report, because you haven′t changed the authority since the last time you ran this report.

The public authority to user profile USRPRFX is changed on January 15, 1994 to *ALL. On February 1, 1994, you run the User Profile Authority Report again. This time, the user profile USRPRF1 will appear on the PRFA1P report as *PUBLIC *ALL. It will also appear on the PRFA2P report as *PUBLIC *ALL.

Example 2: User profile USRPRFX exists on your system and it′s public authority is *EXCLUDE. The public authority has not changed since that last time you ran this report. If you run the User Profile Authority Report on February 1, 1993, it will not list the public authority to user profile USRPRFX as *CHANGE in the PRFA1P report. It will also not list the user profile in the PRFA2P report, because you haven′t changed the authority since the last time you ran this report.

The public authority to user profile USRPRFX is changed on January 15, 1994 to *ALL. On February 1, 1994, you run the User Profile Authority Report again. This time, the user profile USRPRF1 will appear on the PRFA1P report as *PUBLIC *ALL. It will also appear on the PRFA2P report as *PUBLIC *ALL.

Suggestion - Use the PRFA1P and PRFA2P report on a regular basis to review changes to command authority on your report. Unless you have numerous command authority changes, the PRFA2P report should be short and easy to read. It is important to run the PRFA1P regularly to review all user profiles. A user profile which has public authority to it is one of the simplest ways for a hacker to adopt authority once on your system.

15.2.9 Auditor - Job Description Authority Report (JOBDFULL, JOBDDELTA)

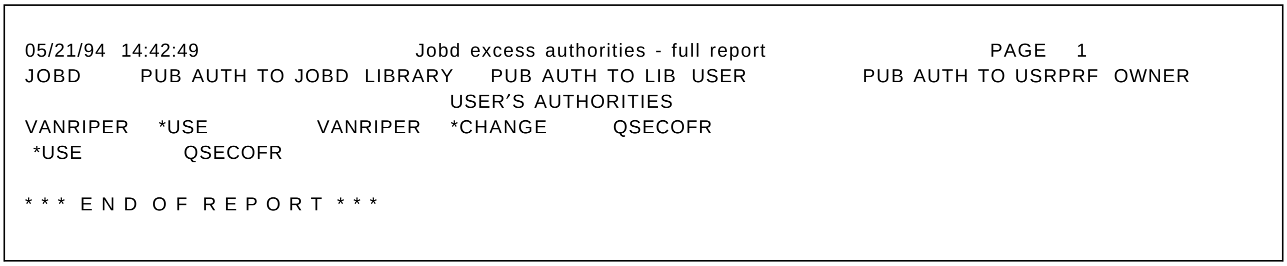
The Job Description Authority Report lists the public authority for the job descriptions on the system.

Note: This is a subsetted list of job descriptions. It generates two reports:

- JOBDFULL - Lists public authority job descriptions
- JOBDDELTA - Lists public authority to job descriptions on the system that have changed since the last time this option was run

This report has been tailored to provide a subset of job descriptions on the system. The intention is to provide a report that will be useful to the reviewer. If all job descriptions are listed, the report tends to become very large and difficult to review. The report has three limiting factors:

- Does NOT include job descriptions that are *PUBLIC *EXCLUDE
- Does NOT include job descriptions which are contained in libraries which are *PUBLIC *EXCLUDE
- Does NOT include job descriptions which contain no user profile, or the user profile listed in the job description has special authorities set to *NONE



```
05/21/94  14:42:49                    Jobd excess authorities - full report                    PAGE    1
JOBD       PUB AUTH TO JOBD  LIBRARY    PUB AUTH TO LIB  USER            PUB AUTH TO USRPRF  OWNER
                                        USER'S AUTHORITIES
VANRIPER    *USE             VANRIPER   *CHANGE        QSECOFR
 *USE          QSECOFR

* * * E N D  O F  R E P O R T * * *
```

Figure 15-5. Sample JOBDFULL Report

Table 15-5. Job Description Report - Report Generated on 02/01/94

Job Description	Library	User profile special Authority	*PUBLIC Authority to Object	*PUBLIC Authority to Library	Last Date Report Run	*PUBLIC Authority Last Change Date	User profile changed special authority	Appear in JOBDFULL	Appear in JOBDDELTA
JOBD1	LIB1	Y	*EXCLUDE	*EXCLUDE	01/01/94	12/15/93	01/01/88	N	N
JOBD2	LIB2	Y	*EXCLUDE	*CHANGE	01/01/94	12/15/93	01/01/88	N	N
JOBD3	LIB1	Y	*CHANGE	*EXCLUDE	01/01/94	12/15/93	01/01/88	N	N
JOBD4	LIB2	Y	*CHANGE	*CHANGE	01/01/94	12/15/93	01/01/88	Y	N
JOBD5	LIB2	Y	*CHANGE	*CHANGE	01/01/94	01/15/94	01/01/88	Y	Y
JOBD6	LIB2	Y	*USE	*CHANGE	01/01/94	12/15/93	01/01/88	Y	Y
JOBD7	LIB2	Y	*ALL	*CHANGE	01/01/94	12/15/93	01/15/94	Y	Y

Example 1: Job description LIBX/JOBDJ exists on your system. The job description contains user USRPRFU which has special authorities defined. Job description JOBDJ′s public authority is *EXCLUDE, and LIBX′s public authority is *EXCLUDE. The public authority has not been changed for either the job description or library since that last time you ran this report. If you run the Job Description Authority Report on January 31, 1994, it will NOT list the job description in either the JOBDFULL or the JOBDDELTA reports because it does not meet the specifications and it has not been changed.

The public authority to either the library or the job description is changed to *CHANGE on February 15. On February 28, 1994, you run the Job Description Authority Report again. This time, the job description LIBX/JOBDJ will appear on the JOBDFULL report as *PUBLIC *CHANGE. It will also appear on the JOBDDELTA report as *PUBLIC *CHANGE. This makes it easy for you to review all the changes to user profiles authority on your system.

Example 2: Job description LIBX/JOBDJ exists on your system. The job description contains user USRPRFU which does NOT have special authorities defined. Job description JOBDJ′s public authority is *CHANGE, and library LIBX′s public authority is *CHANGE. The user profile parameter has not been changed in the job description. You also have not changed the user profile since the last time you ran this report. If you run the Job Description Authority Report on January 31, 1994, it will NOT list job description in either the JOBDFULL or the JOBDDELTA reports because it does not meet the specifications and it has not been changed.

The user profile USRPRFU is changed to contain *ALLOBJ special authority. You run the Job Description Authority Report again. This time, the job description LIBX/JOBDJ will appear on the JOBDFULL report as *PUBLIC *CHANGE. It will also appear on the JOBDDELTA report as *PUBLIC *CHANGE because you changed the user profile USRPRFU.

Suggestion - Use the JOBDDELTA report on a regular basis to review changes to job description authority on your report. Unless you have numerous job description authority changes, the JOBDDELTA report should be short and easy to read. It is important to review the JOBDFULL regularly to review all job descriptions. A job description which has public authority to it is one of the simple ways for a hacker to adopt authority once on your system.

15.2.10 Auditor - Adopted Authority Report (PGMADOPT, PGMDELTA)

The Adopted Authority Report lists the public authority programs which adopt authority on the system.

Note: This is a subsetted list of programs which adopt authority. See accompanying text for more details.

It generates two reports:

- PGMADOPT - Lists public authority to programs which adopt authority
- PGMDELTA - Lists public authority to programs which adopt authority on the system that have changed since the last time this option was run

This report has been tailored to provide a subset of programs which adopt authority on the system. The intention is to provide a report that will be useful to the reviewer. If all programs which adopt authority are listed, the report tends to become very large and difficult to review. The report has three limiting factors:

- Does NOT include programs that adopt authority that are *PUBLIC *EXCLUDE
- Does NOT include programs that adopt authority which are contained in libraries which are *PUBLIC *EXCLUDE
- Does NOT programs that adopt authority whose owner user profile has special authorities set to *NONE

```
User Profile  USRPRF Special Auth
                               Program     PUBLIC ath to PGM  Other Ath Granted  Library     Public ath to LIB
                                 Programs called
 VANRIPER     *ALLOBJ   *JOBCTL   *SAVSYS   *SPLCTL
                               PGM1            *USE                   N          APSSARC         *USE
                                 PGM2       JOBDLIBL3
                               PGM5            *USE                   N                          *USE
                                 PGM3       JOBDLIBL3
                               PGM1            *USE                   N                          *USE
                                 PGM4
     ************ LIBRARY CHANGE ************
                               D600A2          *USE                   N          APSSP010        USER DEF
                                 UNABLE TO RETRIEVE CALLED PROGRAMS
                                 Programs called
```

Figure 15-6. Sample PGMADOPT Report.

Table 15-6. Program Adopt Authority Report - Report Generated on 02/01/94

Program	Library	User Profile Special Authority	*PUBLIC Authority to Program	*PUBLIC Authority to Library	Last Date Report Run	*PUBLIC Authority Last Change Date	User Profile Changed Special Authority	Appear in PGMADOPT	Appear in PGMDELTA
PGM1	LIB1	Y	*EXCLUDE	*EXCLUDE	01/01/94	12/15/93	01/01/88	N	N
PGM2	LIB2	Y	*EXCLUDE	*CHANGE	01/01/94	12/15/93	01/01/88	N	N
PGM3	LIB1	Y	*CHANGE	*EXCLUDE	01/01/94	12/15/93	01/01/88	N	N
PGM4	LIB2	Y	*CHANGE	*CHANGE	01/01/94	12/15/93	01/01/88	Y	N
PGM5	LIB2	Y	*CHANGE	*CHANGE	01/01/94	01/15/94	01/01/88	Y	Y
PGM6	LIB2	Y	*USE	*CHANGE	01/01/94	12/15/93	01/01/88	Y	Y
PGM7	LIB2	Y	*ALL	*CHANGE	01/01/94	12/15/93	01/15/94	Y	Y

Example 1:

If CL program LIBY/PGM1 calls program LIBY/PGM2 (assuming that *PUBLIC has at least *USE authority to the library and programs), then if the command

```
CALL PGM1
```

appears in PGM1 program, the program PGM1 would be listed in the report because the program PGM2 would be executed from the first library in the library list that contained program PGM2. However, if the command

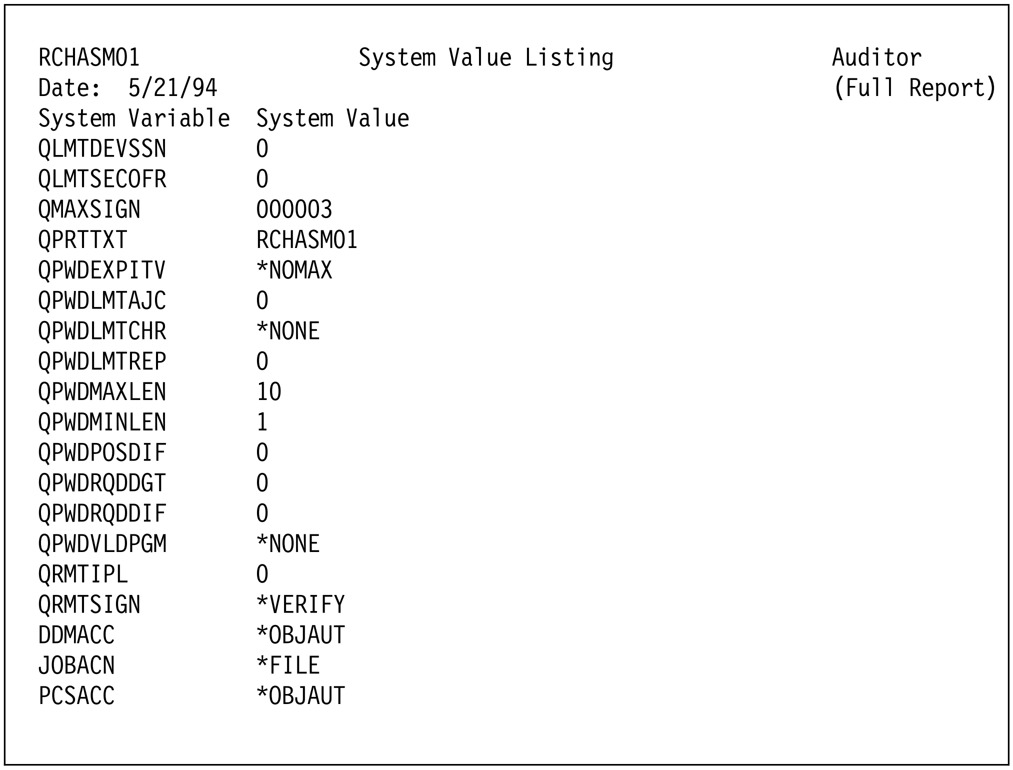
```
CALL LIBx/PGM2
```

appears in the PGM2 program, the program PGM1 would NOT be listed in the report because the program PGM2 can only be executed from library LIB2.

15.2.11 Auditor - System Value/Network Attribute Report (SYSA1P, SYSA2P)

The System Value/Network Attribute Report lists the security related system values and network attributes on the system:

- SYSA1P - Lists all security related system values and network attributes
- SYSA2P - Lists all security related system values and network attributes that have changed since the last time this option was run



```
RCHASM01                System Value Listing                Auditor
Date:  5/21/94                                              (Full Report)
System Variable  System Value
QLMTDEVSSN       0
QLMTSECOFR       0
QMAXSIGN         000003
QPRTTXT          RCHASM01
QPWDEXPITV       *NOMAX
QPWDLMTAJC       0
QPWDLMTCHR       *NONE
QPWDLMTREP       0
QPWDMAXLEN       10
QPWDMINLEN       1
QPWDPOSDIF       0
QPWDRQDDGT       0
QPWDRQDDIF       0
QPWDVLDPGM       *NONE
QRMTIPL          0
QRMTSIGN         *VERIFY
DDMACC           *OBJAUT
JOBACN           *FILE
PCSACC           *OBJAUT
```

Figure 15-7. Sample SYSA1P Report

Table 15-7. System Value/Network Attribute Report - Report

System Value	Last Date Report Run	System Value Last Change Date	Appear in PRFA1P	Appear in PRFA2P
QLMTSECOFR	01/01/94	12/15/93	Y	N
QMAXSIGN	01/01/94	01/15/94	Y	Y

Example: System Value QSECURITY is set to 40 on your system. It has not changed since that last time you ran this report. If you run the System Value/Network Attribute Report on January 1, 1994, it will list the QSECURITY value as ′40′ SYSA1P report. It will not appear on the QSECURITY SYSA2P report, because you haven′t changed it since the last time you ran this report.

QSECURITY is changed from 40 to 30 on January 15, 1994. On February 1, 1994, you run the System Value/Network Attribute report again. This time, QSECURITY will appear on the SYSA1P report as ′30′. It will also appear on the SYSA2P report as ′30′.

Suggestion: Use the SYSA1P and SYSA2P report on a regular basis to review changes to security-related system values and network attributes on your system. Unless you have numerous changes to system values and network attributes authority changes, the both the SYSA1P and SYSA2P reports should be short and easy to read. It is important to run the SYSA1P regularly to review all security related system values and network attributes. System values and network attributes are very sensitive and can severely compromise your security if not properly set.

15.2.12 Auditor - Authorization List Report (AUTA1P, AUTA2P)

The Authorization List Report lists all authorization lists on the system and their contents. It generates two reports:

- AUTA1P - Lists all authorization lists on the system and their contents
- USRA2P - Lists all authorization lists on the system and their contents that have changed since the last time this option was run

```
RCHASM01            Authorization List Contents and Authorizations    Auditor
Date:  5/21/94                                                        (Full Report)
Library    Auth List  Owner      User      Authority List mgt Oper Mgt Read Add Updt Delt
QSYS       AUTLCHG    EVANS      EVANS      *ALL        X       X   X    X    X    X   X    X
QSYS       AUTLCHG    EVANS      *PUBLIC    *CHANGE             X             X    X   X    X
QSYS       AUTLLESS   EVANS      EVANS      *ALL        X       X   X    X    X    X   X    X
QSYS       AUTLLESS   EVANS      QUSER      *USE                X             X
QSYS       AUTLLESS   EVANS      *PUBLIC    *CHANGE             X             X    X   X    X
QSYS       AUTLMORE   EVANS      EVANS      *ALL        X       X   X    X    X    X   X    X
QSYS       AUTLMORE   EVANS      QUSER      *ALL                X   X    X    X    X   X    X
QSYS       AUTLMORE   EVANS      *PUBLIC    *CHANGE             X             X    X   X    X
QSYS       AUTL1      ITSCID01   ITSCID01   *ALL        X       X   X    X    X    X   X    X
QSYS       AUTL1      ITSCID01   EVANS      *CHANGE     X       X             X    X   X    X
QSYS       AUTL1      ITSCID01   ELLEN      *USE                X             X
QSYS       AUTL1      ITSCID01   GRPPRF1    *CHANGE             X             X    X   X    X
QSYS       AUTL1      ITSCID01   NEWUSER    *USE                X             X
QSYS       AUTL1      ITSCID01   *PUBLIC    *EXCLUDE
QSYS       JANZEN     GROUP      GROUP      USER DEF    X
QSYS       JANZEN     GROUP      MAGGIE     *USE                X             X
QSYS       JANZEN     GROUP      SERGIO     *CHANGE             X             X    X   X    X
QSYS       JANZEN     GROUP      *PUBLIC    *EXCLUDE
QSYS       QCQRPSAUTL QSYS       QSYS       *ALL        X       X   X    X    X    X   X    X
QSYS       QCQRPSAUTL QSYS       *PUBLIC    *EXCLUDE
QSYS       QFMCAUTL   QSYS       QSYS       *ALL        X       X   X    X    X    X   X    X
QSYS       QFMCAUTL   QSYS       QSECOFR    *CHANGE             X             X    X   X    X
QSYS       QFMCAUTL   QSYS       QOSIFS     USER DEF    X
QSYS       QFMCAUTL   QSYS       KRIS       *CHANGE             X             X    X   X    X
QSYS       QFMCAUTL   QSYS       *PUBLIC    *EXCLUDE
QSYS       QIWSADM    QSECOFR    QSECOFR    *ALL        X       X   X    X    X    X   X    X
QSYS       QIWSADM    QSECOFR    *PUBLIC    *USE                X             X
QSYS       QRTL       QSYS       QSYS       *ALL        X       X   X    X    X    X   X    X
QSYS       QRTL       QSYS       QRTL       *CHANGE             X             X    X   X    X
QSYS       QRTL       QSYS       *PUBLIC    *CHANGE             X             X    X   X    X
QSYS       TAACVTQHST QSECOFR    QSECOFR    *ALL        X       X   X    X    X    X   X    X
QSYS       TAACVTQHST QSECOFR    *PUBLIC    *EXCLUDE
QSYS       TAADLTQHST QSECOFR    QSECOFR    *ALL        X       X   X    X    X    X   X    X
QSYS       TAADLTQHST QSECOFR    *PUBLIC    *EXCLUDE
QSYS       TAADSPADP  XVT1925    XVT1925    *ALL        X       X   X    X    X    X   X    X
QSYS       TAADSPADP  XVT1925    EVANS      *CHANGE             X             X    X   X    X
QSYS       TAADSPADP  XVT1925    *PUBLIC    *EXCLUDE
```

Figure 15-8. Sample AUTA1P Report

Table 15-8. Authorization List Report - Report Generated on 02/01/94

Authorization List	Last Date Report Run	Authorization Last Change Date	Appear in PRFA1P	Appear in PRFA2P
VANRIPER	01/01/94	12/15/93	Y	N
VANRIPER2	01/01/94	01/15/94	Y	Y

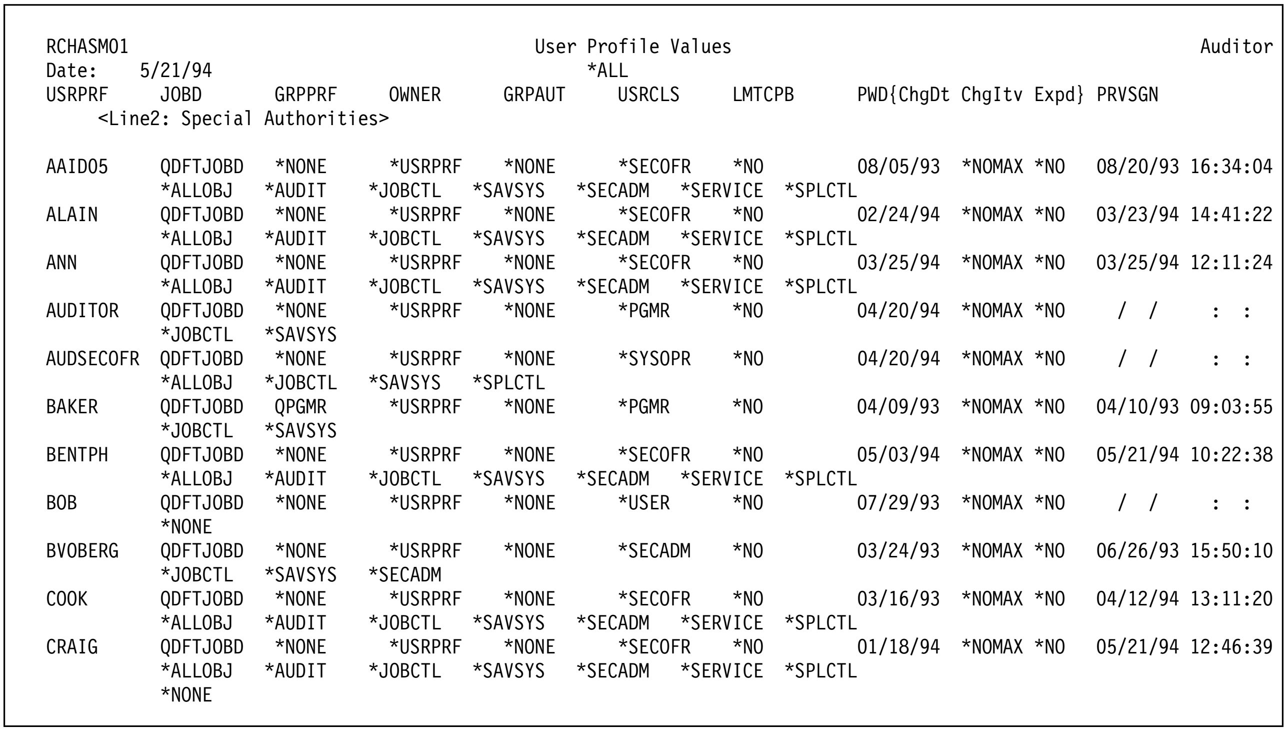
Example: Authorization List AUTLL exists on your system and it contains user profile USRPRFU. USRPRFU authority level is set to *EXCLUDE. The authority has not changed since that last time you ran this report. If you run the Authorization List Report on February 1, 1994, it will list authorization list AUTLL and all its contents in the AUTA1P report. It will not list the authorization list in the AUTA2P report, because you haven′t changed the authorization list since the last time you ran this report.

The authorization list AUTLL is changed so that user profile USRPRFU has authority level *CHANGE on January 15, 1994 to *PUBLIC *ALL. On February 1, 1994, you run the Authorization List Report again. This time, the authorization list AUTLL and all its contents will appear on the AUTA1P. The authorization list and the user profile USRPRFU will appear on the USRA2P report. However, not all contents of the authorization list will appear on the USRA2P report.

Suggestion - Use the AUTA2P report on a regular basis to review changes to authorization lists on your system. Unless you have numerous authorization lists on your system, this report should be easy to read and review. Use the AUTA1P in conjunction with other reports to perform in depth reviews of security on your system.

15.2.13 Auditor - List All User Profiles (USRA1P)

The List All User Profiles report lists basic information for all users on the system. It generates only 1 report, USRA1P.



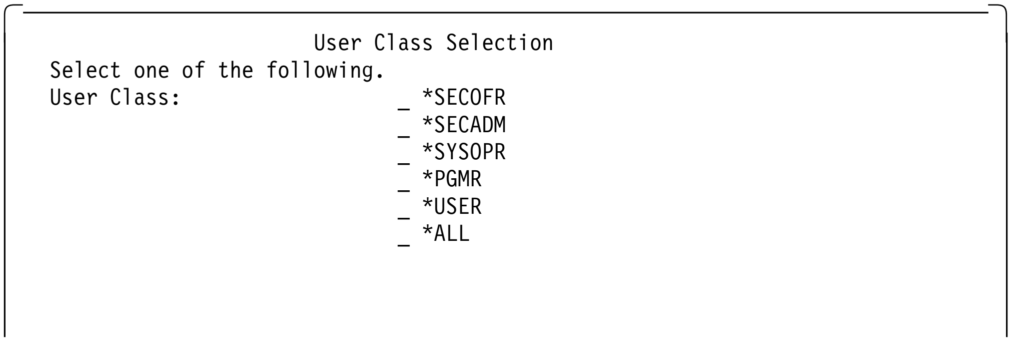
```
RCHASMO1                                User Profile Values                                       Auditor
Date:    5/21/94                             *ALL
USRPRF     JOBD       GRPPRF     OWNER      GRPAUT     USRCLS     LMTCPB      PWD{ChgDt ChgItv Expd} PRVSGN
     <Line2: Special Authorities>

AAIDO5     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         08/05/93  *NOMAX *NO   08/20/93 16:34:04
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ALAIN      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         02/24/94  *NOMAX *NO   03/23/94 14:41:22
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ANN        QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/25/94  *NOMAX *NO   03/25/94 12:11:24
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
AUDITOR    QDFTJOBD   *NONE      *USRPRF    *NONE      *PGMR      *NO         04/20/94  *NOMAX *NO     /  /      :  :
           *JOBCTL   *SAVSYS
AUDSECOFR  QDFTJOBD   *NONE      *USRPRF    *NONE      *SYSOPR    *NO         04/20/94  *NOMAX *NO     /  /      :  :
           *ALLOBJ   *JOBCTL   *SAVSYS   *SPLCTL
BAKER      QDFTJOBD   QPGMR      *USRPRF    *NONE      *PGMR      *NO         04/09/93  *NOMAX *NO   04/10/93 09:03:55
           *JOBCTL   *SAVSYS
BENTPH     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         05/03/94  *NOMAX *NO   05/21/94 10:22:38
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
BOB        QDFTJOBD   *NONE      *USRPRF    *NONE      *USER      *NO         07/29/93  *NOMAX *NO     /  /      :  :
           *NONE
BVOBERG    QDFTJOBD   *NONE      *USRPRF    *NONE      *SECADM    *NO         03/24/93  *NOMAX *NO   06/26/93 15:50:10
           *JOBCTL   *SAVSYS   *SECADM
COOK       QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/16/93  *NOMAX *NO   04/12/94 13:11:20
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
CRAIG      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         01/18/94  *NOMAX *NO   05/21/94 12:46:39
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
           *NONE
```

Figure 15-9. Sample USRA1P Report

Suggestion - Use the USRA1P to periodically to review all user profiles on your system. This is an excellent way to find user profiles that have too much authority or have not signed on recently.

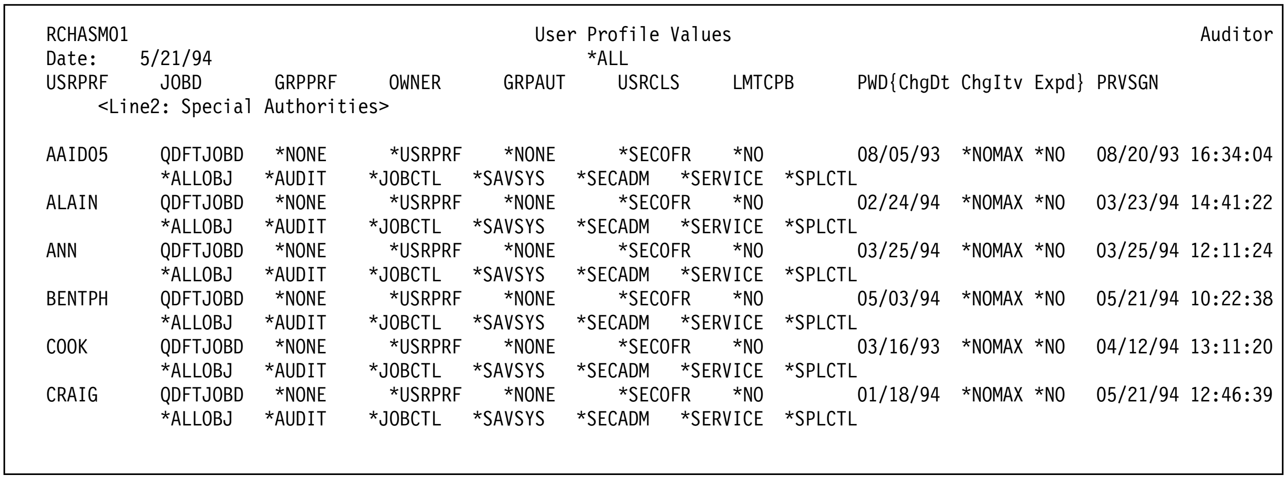
15.2.14 Auditor - List All User Profiles by User Class (USRA1P)

The List All User Profiles by User Class report lists basic information for all users in a specific user class on the system. It generates only one report, USRA1P. When this option is selected, an additional panel is displayed to select which user class should appear on the report. See Chapter 3, “User Profiles and Group Profiles” on page 3-1 for further details.



```
                         User Class Selection
Select one of the following.
User Class:                _ *SECOFR
                           _ *SECADM
                           _ *SYSOPR
                           _ *PGMR
                           _ *USER
                           _ *ALL
```

User Class Selection



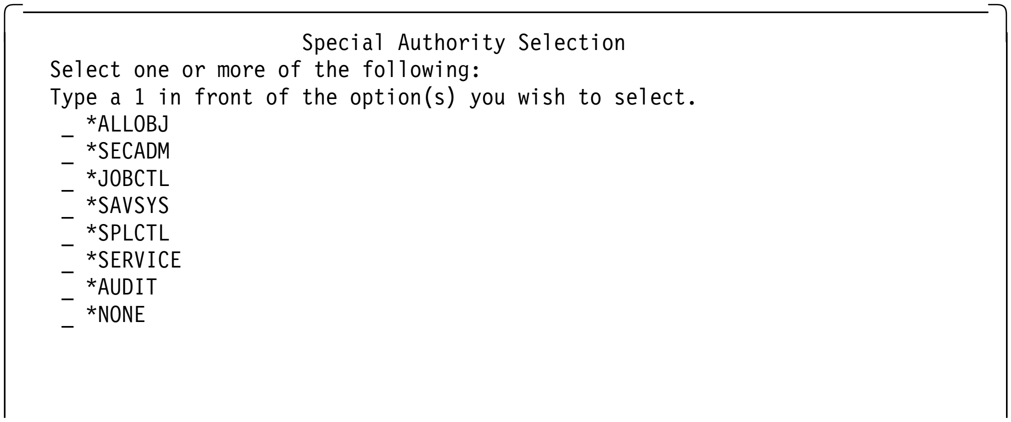
```
RCHASM01                                   User Profile Values                                      Auditor
Date:    5/21/94                               *ALL
USRPRF     JOBD       GRPPRF     OWNER      GRPAUT     USRCLS     LMTCPB      PWD{ChgDt ChgItv Expd} PRVSGN
     <Line2: Special Authorities>

AAID05     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         08/05/93  *NOMAX *NO   08/20/93 16:34:04
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ALAIN      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         02/24/94  *NOMAX *NO   03/23/94 14:41:22
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ANN        QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/25/94  *NOMAX *NO   03/25/94 12:11:24
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
BENTPH     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         05/03/94  *NOMAX *NO   05/21/94 10:22:38
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
COOK       QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/16/93  *NOMAX *NO   04/12/94 13:11:20
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
CRAIG      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         01/18/94  *NOMAX *NO   05/21/94 12:46:39
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
```

Figure 15-10. Sample USRA2P Report

Suggestion - Use the USRA1P periodically to review all user profiles in a specific user class on your system. This is an excellent way to find user profiles that have too much authority.

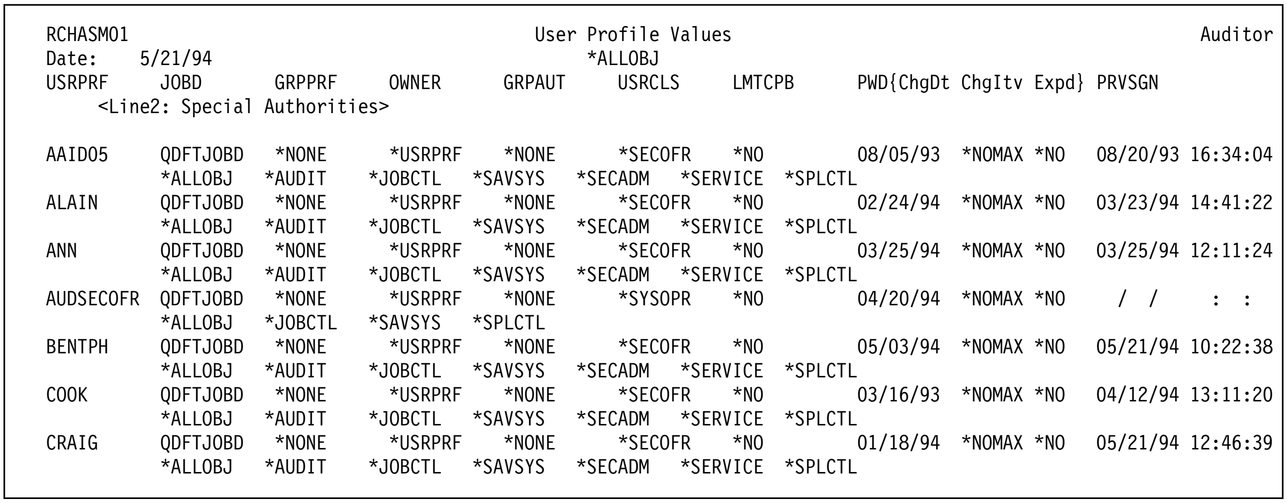
15.2.15 Auditor - List All User Profiles by Special Authority (USRA1P)

The List All User Profiles by Special Authority report lists basic information about all users who have a specific special authority. It generates only one report, USRA1P. When this option is selected, an additional panel is displayed to select which special authorities should appear on the report.



```
                         Special Authority Selection
   Select one or more of the following:
   Type a 1 in front of the option(s) you wish to select.
    _ *ALLOBJ
    _ *SECADM
    _ *JOBCTL
    _ *SAVSYS
    _ *SPLCTL
    _ *SERVICE
    _ *AUDIT
    _ *NONE
```

You may select more than one class to appear on the report, but you may have duplicate entries. For instance, if you select to have a report generated to have all users with *ALLOBJ or *SECADM special authority, you will receive a report which contains 2 sections. The first section will contain all user profiles with *ALLOBJ special authority. The second section will contain all user profiles with *SECADM special authority. The user profiles which have both *ALLOBJ and *SECADM authority will be listed in the first section of the report as well as the second section of the report.



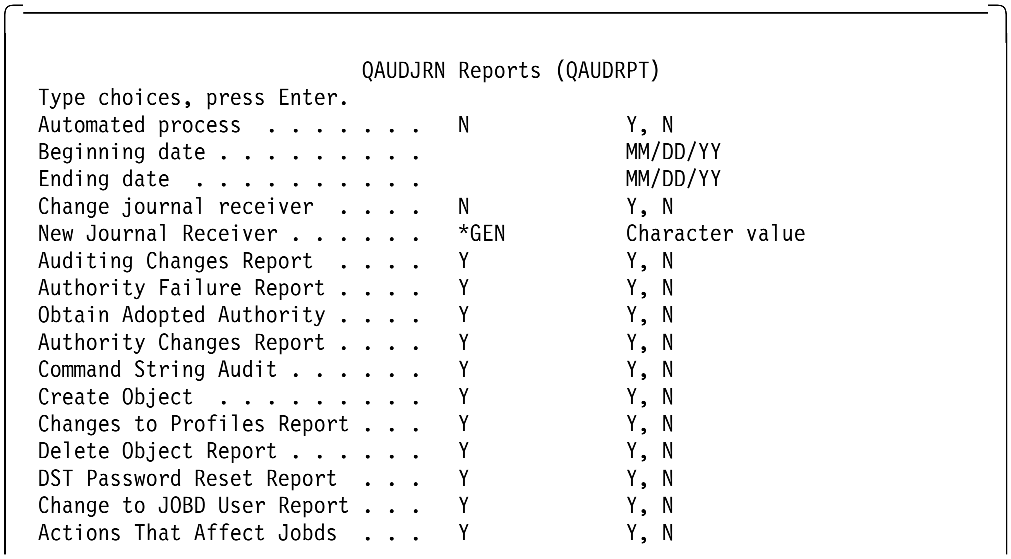
```
RCHASM01                                    User Profile Values                                   Auditor
Date:    5/21/94                                 *ALLOBJ
USRPRF     JOBD       GRPPRF     OWNER      GRPAUT     USRCLS     LMTCPB      PWD{ChgDt ChgItv Expd} PRVSGN
     <Line2: Special Authorities>

AAID05     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         08/05/93  *NOMAX *NO   08/20/93 16:34:04
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ALAIN      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         02/24/94  *NOMAX *NO   03/23/94 14:41:22
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
ANN        QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/25/94  *NOMAX *NO   03/25/94 12:11:24
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
AUDSECOFR  QDFTJOBD   *NONE      *USRPRF    *NONE      *SYSOPR    *NO         04/20/94  *NOMAX *NO     /  /      :  :
           *ALLOBJ   *JOBCTL   *SAVSYS   *SPLCTL
BENTPH     QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         05/03/94  *NOMAX *NO   05/21/94 10:22:38
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
COOK       QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         03/16/93  *NOMAX *NO   04/12/94 13:11:20
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
CRAIG      QDFTJOBD   *NONE      *USRPRF    *NONE      *SECOFR    *NO         01/18/94  *NOMAX *NO   05/21/94 12:46:39
           *ALLOBJ   *AUDIT    *JOBCTL   *SAVSYS   *SECADM   *SERVICE  *SPLCTL
```

Figure 15-11. Sample USRA3P Report

Suggestion - Use the USRA1P periodically to review all user profiles which have a specific special authority on your system. This is an excellent way to find user profiles that have too much authority.

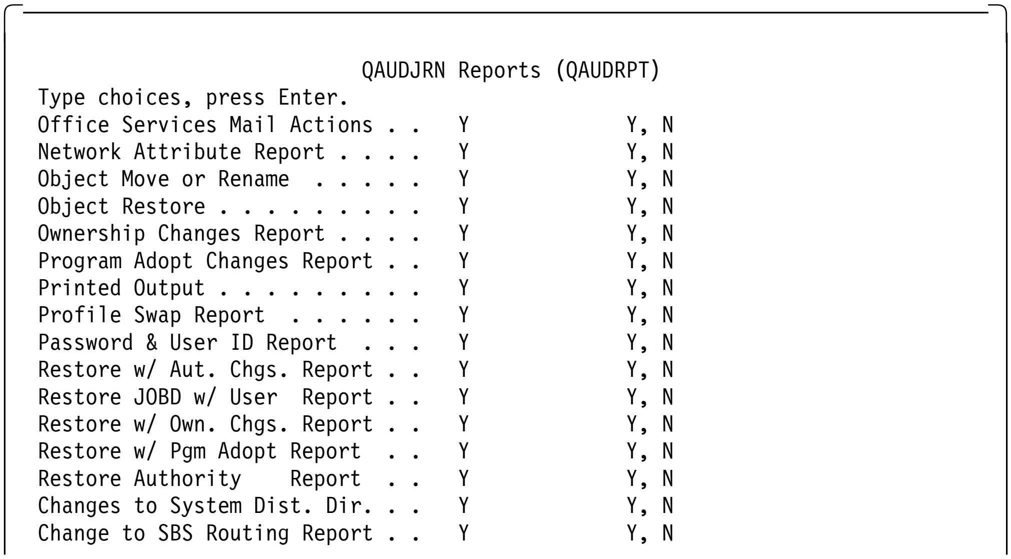
15.2.16 Auditor - QAUDJRN Reports (QPQPRFIL)

This option allows you to generate reports from the system security journal, QAUDJRN. It uses the journal type entry to create an individual report for each of the journal types. The entries in the reports are taken directly from the audit journal.



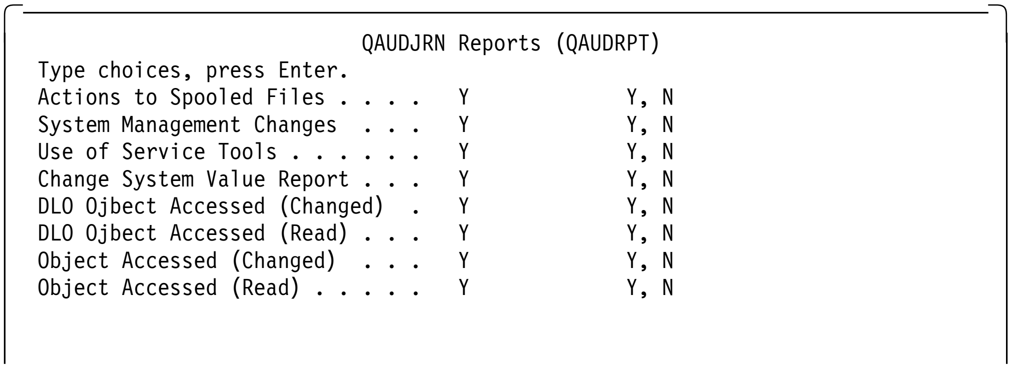
```
                              QAUDJRN Reports (QAUDRPT)
Type choices, press Enter.
Automated process  . . . . . . .   N              Y, N
Beginning date . . . . . . . . .                  MM/DD/YY
Ending date  . . . . . . . . . .                  MM/DD/YY
Change journal receiver  . . . .   N              Y, N
New Journal Receiver . . . . . .   *GEN           Character value
Auditing Changes Report  . . . .   Y              Y, N
Authority Failure Report . . . .   Y              Y, N
Obtain Adopted Authority . . . .   Y              Y, N
Authority Changes Report . . . .   Y              Y, N
Command String Audit . . . . . .   Y              Y, N
Create Object  . . . . . . . . .   Y              Y, N
Changes to Profiles Report . . .   Y              Y, N
Delete Object Report . . . . . .   Y              Y, N
DST Password Reset Report  . . .   Y              Y, N
Change to JOBD User Report . . .   Y              Y, N
Actions That Affect Jobds  . . .   Y              Y, N
```

Screen 1



```
                           QAUDJRN Reports (QAUDRPT)
Type choices, press Enter.
Office Services Mail Actions . .   Y             Y, N
Network Attribute Report . . . .   Y             Y, N
Object Move or Rename  . . . . .   Y             Y, N
Object Restore . . . . . . . . .   Y             Y, N
Ownership Changes Report . . . .   Y             Y, N
Program Adopt Changes Report . .   Y             Y, N
Printed Output . . . . . . . . .   Y             Y, N
Profile Swap Report  . . . . . .   Y             Y, N
Password & User ID Report  . . .   Y             Y, N
Restore w/ Aut. Chgs. Report . .   Y             Y, N
Restore JOBD w/ User  Report . .   Y             Y, N
Restore w/ Own. Chgs. Report . .   Y             Y, N
Restore w/ Pgm Adopt Report  . .   Y             Y, N
Restore Authority    Report  . .   Y             Y, N
Changes to System Dist. Dir. . .   Y             Y, N
Change to SBS Routing Report . .   Y             Y, N
```

Screen 2



```
                           QAUDJRN Reports (QAUDRPT)
Type choices, press Enter.
Actions to Spooled Files . . . .   Y             Y, N
System Management Changes  . . .   Y             Y, N
Use of Service Tools . . . . . .   Y             Y, N
Change System Value Report . . .   Y             Y, N
DLO Ojbect Accessed (Changed)  .   Y             Y, N
DLO Ojbect Accessed (Read) . . .   Y             Y, N
Object Accessed (Changed)  . . .   Y             Y, N
Object Accessed (Read) . . . . .   Y             Y, N
```

Screen 3

This option is actually a command named QAUDRPT. You can call the command directly, or be using the menu.

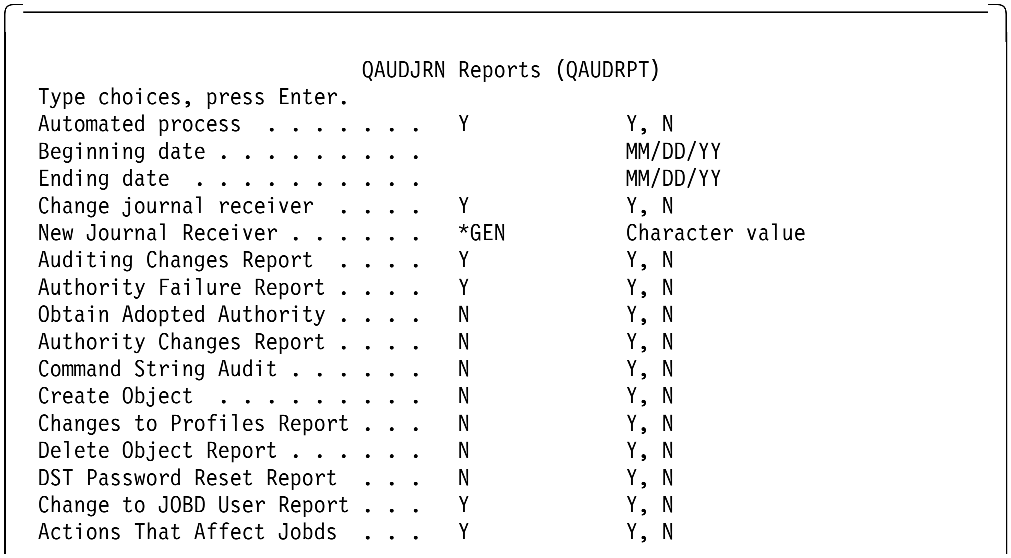
This application allows you to select a date range against which to run the reports. For instance, if you want to run a monthly report for all options for the month of May, 1993, you would enter the Beginning date as 05/01/94 and the Ending date.

Note: If you specify an invalid date range you will get unexpected results. Also, the journal receivers for the entire date range must be present on the system, or you will get unexpected results.

If you only want to generate a report based on the current journal receiver, do not specify a date range, and change the Automate Process parameter from N to Y. This will use only the current journal to generate a report.

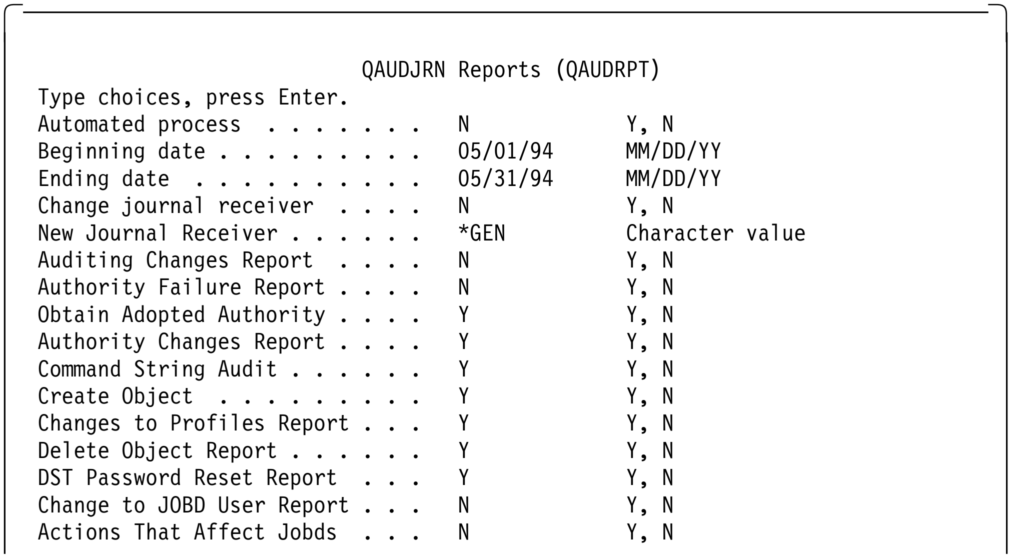
If you want the journal receiver to be changed at the time the report is run, change the Change Journal Receiver parameter from N to Y. This will change the journal receiver before the report is run. If you choose to change the journal receiver at the same time, you can also choose to name the new journal receiver. The default is to let the system generate the new journal receiver name.

Suggestion: Run daily or weekly reports for pertinent information relative to your environment. Choose to automate the process and generate a new journal receiver. Do not put in a date range and schedule this on the system job scheduler to run at the frequency you want it to run. Use a scenario similar to:



```
                           QAUDJRN Reports (QAUDRPT)
Type choices, press Enter.
Automated process  . . . . . . .   Y              Y, N
Beginning date . . . . . . . . .                  MM/DD/YY
Ending date  . . . . . . . . . .                  MM/DD/YY
Change journal receiver  . . . .   Y              Y, N
New Journal Receiver . . . . . .   *GEN           Character value
Auditing Changes Report  . . . .   Y              Y, N
Authority Failure Report . . . .   Y              Y, N
Obtain Adopted Authority . . . .   N              Y, N
Authority Changes Report . . . .   N              Y, N
Command String Audit . . . . . .   N              Y, N
Create Object  . . . . . . . . .   N              Y, N
Changes to Profiles Report . . .   N              Y, N
Delete Object Report . . . . . .   N              Y, N
DST Password Reset Report  . . .   N              Y, N
Change to JOBD User Report . . .   Y              Y, N
Actions That Affect Jobds  . . .   Y              Y, N
```

Suggestion: Run monthly reports for non-critical information relative to your environment. Do not choose to automate the process and generate a new journal receiver. Put in a date range and schedule and submit the job to run or put an entry on the system job scheduler to run on the date you want it to run. Use a scenario similar to:



```
                           QAUDJRN Reports (QAUDRPT)
Type choices, press Enter.
Automated process  . . . . . . .   N              Y, N
Beginning date . . . . . . . . .   05/01/94       MM/DD/YY
Ending date  . . . . . . . . . .   05/31/94       MM/DD/YY
Change journal receiver  . . . .   N              Y, N
New Journal Receiver . . . . . .   *GEN           Character value
Auditing Changes Report  . . . .   N              Y, N
Authority Failure Report . . . .   N              Y, N
Obtain Adopted Authority . . . .   Y              Y, N
Authority Changes Report . . . .   Y              Y, N
Command String Audit . . . . . .   Y              Y, N
Create Object  . . . . . . . . .   Y              Y, N
Changes to Profiles Report . . .   Y              Y, N
Delete Object Report . . . . . .   Y              Y, N
DST Password Reset Report  . . .   Y              Y, N
Change to JOBD User Report . . .   N              Y, N
Actions That Affect Jobds  . . .   N              Y, N
```

This single option may generate a large number of reports. The default shipped with the IBM Security/400 application is to produce all reports. You can indicate that you do not want to have a report produced by changing the ′Y′ to a ′N′. The following table indicates which values are used as criteria for each report.

Table 15-9. Reports Produced from QAUDJRN Based on Journal Entry Type

Report Title	**Selected QAUDJRN Records**
Auditing Changes	AD
Authority Failure	AF
Obtain Adopted Authority	AP
Authority Changes Report	CA
Command String Audit	CD
Create Object	CO
Changes to Profiles	CP
Delete Object	DO
DST Password Reset	DS
Change to JOBD User	JD
Actions That Affect Jobds	JS
Office Services Mail Actions	ML
Network Attribute	NA
Object Move or Rename	OM
Object Restore	OR
Ownership Changes	OW
Program Adopt Changes	PA
Printed Output	PO
Profile Swap	PS
Password and User ID	PW
Restore with Authority Changes	RA
Restore JOBD with User Profile Specified	RJ
Restore with Ownership Changes	RO
Restore with Program Adoption	RP
Restore Authority	RU
Changes to System Distribution Directory	SD
Change to SBS Routing	SE
Actions to Spooled Files	SF
System Management Changes	SM
Use of Service Tools	ST
Change System Value	SV
DLO Object Accessed (Changed)	YC
DLO Object Accessed (Read)	YR
Ojbect Accessed (Changed)	ZC
Ojbect Accessed (Read)	ZR

All reports generated from this option are generated using Query/400. Therefore, all reports generated with this option are generated with the name QPQPRFIL.

You will most likely want to tailor the reports for your individual needs. Please see 15.3, "Tailoring IBM Security/400" on page 15-25 for more details on how to tailor these reports.

15.2.17 General Performance Considerations

Auditor

The applications included in Auditor portion of IBM Security/400 are by default submitted for batch processing. Many of the applications interrogate the entire system to gather the needed information. For instance, the File Authority Report interrogates each and every file on your system. The application uses the DSPOBJD command to generate a list of all files on the system. If you have a large number of files, this task will take several hours to complete. You should consider performance conflicts with any other batch applications you may be running. If you are running your batch subsystem as a single threaded subsystem (maximum jobs = 1), your batch subsystem may become dedicated solely to processing the IBM Security/400 applications. If your environment can tolerate a multi-threaded batch subsystem, you should consider changing the maximum jobs to a value greater than one. The following applications supplied with IBM Security/400 can be significantly long running:

- File authority
- Library authority
- Command authority
- Job description authority
- Adopted authority

Suggestion: Submit the long running batch jobs during a lull period. If you are using the reports to perform an audit, ask that the auditee submit the jobs to batch before you arrive, or during the night. If you run this during regular work hours, you may impact the performance of the system′s batch and interactive environment.

Remove Inactive User

The Remove Inactive User application has been tested and found not to have a significant impact on performance.

Use SECADM Profile

The Use SECADM Profile application has been tested and found not to have a significant impact on performance.

Universal Access *NONE

The Universal Access *NONE application has been tested and found to have a slight impact on batch performance. This application interrogates all libraries and root level folders on your system to identify libraries and root level folders that are improperly secured.

Suggestion - The Universal Access *NONE application should be scheduled when your system does not have significant batch activity scheduled.

Systematic Logon Reporting

The Systematic Logon Detection and Notification application may have performance impacts. This application is designed to be run in a batch subsystem and remain active. It will wake up periodically and interrogate the system audit journal, QAUDJRN. The default period of time shipped with the application is 1 hour. However, you may change this to any length of time that you desire. If you shorten the period of time, you should consider performance impacts. The application will copy the entries in the QAUDJRN to a physical file to be queried. If you have a large audit journal, or you set the wake up period to a relatively short amount of time (10 minutes or less), the performance of your batch environment may be severely degraded.

Suggestion: If you are concerned with security violations on your system, you should implement this application. If you are comfortable that you are detecting security violations in a timely manner, you may consider not implementing this application.

15.3 Tailoring IBM Security/400

All source code for IBM Security/400 applications are shipped with the product and are stored in source files in two libraries:

- APSSNLS - Contains source code for:
  - Remove Inactive User
  - Use SECADM Profile
  - Universal Access *NONE
  - Systematic Logon Reporting
- AUDITOR - Contains source code for all Auditor reports

Source code is located in the following files in library APSSNLS:

- QCLSRC
- QCMDSRC
- QRPGSRC
- QDDSSRC

15.3.1 Remove Inactive User

Two items of information are easily tailored:

- The user id and address of your security administrator are stored in a data area, DBDA4620. This information is used to communicate between the application and your security administrator. To change the data area, enter the following command:

  ```
  CHGDTAARA DTAARA(DBDA4620) NEWVALUE(VANRIPER RCHAS1)
  ```

  In this case, the user ID of your security administrator is VANRIPER and the address is RCHAS1.

- The application also allows you to specify user profiles that should be exclude from the Remove Inactive User application. This means that even if they haven′t signed on, they still will not be deleted. These names are stored in a file named DBFPIU70. To edit this file, use the DFU utility to create a temporary program to update the file.

15.3.2 Use SECADM Profile

You should consider who should be authorized to access this application. All objects associated with this application should be *PUBLIC *EXCLUDE.

15.3.3 Use Universal Access *NONE

This application is designed to secure libraries and root level folders from public access. However, many of the system libraries and folders must have public access for your system to run properly. All files and folders that require public access are stored in a file DBFPAO10. It is shipped with system supplied libraries and folders for language 2924 at V2R2. You may have to update this file to contain your unique system libraries and folders. In addition, your system may have other libraries and folders that public must have access to. These libraries and folders may also be stored in this file. To edit this file, use the DFU utility to create a temporary program to update the file.

15.3.4 Systematic Logon Reporting

Source code is located in one file: APSSNLS/QCLSRC.

Two data areas are provided for customization:

- The user ID and address of your security administrator are stored in a data area, DBDA4620. This information is used to communicate between the application and your security administrator. To change the data area, enter the following command:

  ```
  CHGDTAARA DTAARA(DBDA4620) NEWVALUE(VANRIPER RCHAS1)
  ```

  In this case, the user ID of your security administrator is VANRIPER and the address is RCHAS1.

- The second data area is used to identify threshold limits that are used to identify when the security administrator should be notified of excessive signon attempts. Two thresholds are used:

  - Number of invalid signon attempts since 12:00 AM: This number represents the number of invalid signon attempts that are acceptable. For instance, if you want to be notified when the number of invalid signon attempts reaches 50, you would specify your threshold as 50. If during the day, any combination of invalid signon attempts reaches 50, your security administrator will be notified.

  - Number of user profile disables since 12:00 AM: This number represents the number of user profile disables that are acceptable on a daily basis. For instance, if you want to be notified when the number of user profile disables reaches 10, you would specify your threshold as 10. If during the day, 10 user profiles are disabled, your security administrator will be notified.

  The data area is broken into two parts, one part for the invalid signon attempts and the other part for the number of disables. You should determine what your thresholds should be and enter them into the data area. For example:

  ```
  CHGDTAARA DTAARA(DBDA2270) NEWVALUE(5010)
  ```

  Positions 1-2 contain the number of invalid signon attempts allowed before notification. Positions 3-4 contain the number of user profile disables allowed before notification.

15.3.5 Auditor

Auditor source code is also located in an additional file, AUDITOR/QMENUSRC.

Many queries are used in AUDITOR. The queries all start with the naming convention D519* and have a description associated with them to assist you in locating the queries you wish to run. These queries can be tailored to your environment. The defaults have pertinent information for the unique environment under which IBM Security/400 was developed. You can easily update the queries using the Query/400 Licensed Program Product.

Two of the applications generate reports that subset the lists of objects on the system. These two applications are:

- Job Description Authority
- Adopted Authority

You may want to change these applications to produce a full report, rather than a subsetted list.

The remaining applications generate full lists of objects. You may want to change the application to provide a subsetted report. Many of the applications generate reports directly from RPG, others use Query/400.

To update an any source associated with an application, refer to the IBM Security/400 technical information supplied with the product.

15.3.6 Translation Considerations

The source code in APSSNLS should be fairly easy to translate. The code was designed to be NLS enabled so that you can easily change the message files associated with the applications to your own language. If you do not wish to modify the message files you may modify the source code to translate the code to your own language.

The source code in Auditor is not NLS enabled. To translate it to your own language, you will need to modify the menus, display files, message files and reports.

Appendix A. QSYSMSG Message Queue

The QSYSMSG message queue is an optional message queue that you can create in QSYS library to direct special system messages to it. In this way, you can write a program to gain control when one of the special system messages arrives at QSYSMSG message queue. This program should be written as a break-handling program.

Note: A break-handling program is one that is automatically called when a message arrives at a message queue (in this case at QSYSMSG) that is in *BREAK mode. The name of the program and break delivery must both be specified on the same Change Message Queue (CHGMSGQ) command. To find out more about break-handling programs refer to the *AS/400 Programming: Control Language's Programmers Guide*.

Note that some messages directed to QSYSMSG are also directed to QSYSOPR message queue, but some others are not.

Enter the following command to create the QSYSMSG message queue:

```
CRTMSGQ QSYS/QSYSMSG +
        TEXT('Optional message queue to receive important messages')
```

You can find all the specific messages that are directed to the QSYSMSG message queue and examples of monitoring programs in the *AS/400 Programming: Control Language Programmer's Guide*.

From a security point of view, the following messages are important:

CPF1269 `Program start request received on communications device was rejected with reason codes.`

This message is sent when a start request is rejected and contains a reason code identifying why the rejection occurred. For a detailed explanation of each reason code, refer to the *AS/400: ICF Programmer's Guide*.

A start request could be rejected because of an invalid password or an unauthorized condition (probably to a device description). It may mean that someone is trying to break your system's security. You can do the following:

Establish a threshold for the number of attempts of invalid requests in a given period. If the threshold is reached (which can be monitored by a program) you can:

- Send a message to the QSYSOPR message
- Record the attempt for the security officer to review
- Issue the End Mode (ENDMOD) command to set the allowed jobs to zero, if the problem is in an APPC communication (included PC Support). This allows jobs currently using the device description to remain active, but prevents other jobs to start a request.

CPI1393 `Subsystem xxxx disabled user profile yyyy on device zzzz.`

This message is sent when the user reaches the maximum number of signon attempts, which is determined in the

QMAXSIGN system value. The action taken when that value is reached is specified in the QMAXSGNACN system value.

QSYSMSG Receiving Program Example

The following is a sample program which receives messages from QSYSMSG message queue. The program will handle the message CPF1269. The reason codes for that message are in binary so it is necessary to convert them to decimal for the comparisons. The only reason codes that it controls are 704 and 705. If one of this conditions are met, the program send a message to QSECOFR message queue, if not just send a message to QSYSOPR message queue.

You should use a separate job to call this program (using SBMJOB) and the program will remain active, waiting for a message to arrive, until someone ends it with the ENDJOB command.

As we mentioned, this is just a sample program. You can modify its complexity to control more conditions and events.

```
PGM
/******************************************************************/
/* SIMPLE PROGRAM THAT MONITORS QSYSMSG MESSAGE QUEUE             */
/*                                                                */
/* THE PROGRAM LOOKS FOR THE MESSAGE CPF1269 WITH REASON          */
/* CODE 704 AND 705.  IF THEY ARE FOUND, SENDS A MESSAGE TO       */
/* QSECOFR REPORTING THE FAILURE, OTHERWISE SENDS A MESSAGE       */
/* TO QSYSOPR MESSAGE QUEUE.                                      */
/******************************************************************/

             DCL        VAR(&MSG) TYPE(*CHAR) LEN(132)
             DCL        VAR(&MSGID) TYPE(*CHAR) LEN(7)
             DCL        VAR(&MSGDTA) TYPE(*CHAR) LEN(100)
             DCL        VAR(&DEVICE) TYPE(*CHAR) LEN(10)

/* RECEIVE THE MESSAGE AND ITS DATA                               */


START:       RCVMSG     MSGQ(QSYS/QSYSMSG) WAIT(*MAX) MSG(&MSG) +
                          MSGDTA(&MSGDTA) MSGID(&MSGID)

             IF         COND(&MSGID *EQ ’CPF1269’)
              IF         COND((%BIN(&MSGDTA 45 2) *EQ 704) *OR +
                           (%BIN(&MSGDTA 45 2) *EQ 705)) THEN(DO)
                CHGVAR     VAR(&DEVICE) VALUE(%SST(&MSGDTA 1 10))
                         /* Extract device name */
                SNDPGMMSG  MSG(’Device ’ *CAT &DEVICE *TCAT ’ had a +
                            security failure’) TOMSGQ(QSECOFR)
              ENDDO
              ELSE
/* IF IT IS ANOTHER TYPE OF MESSAGE, JUST SEND TO QSYSOPR MSGQ    */

              SNDPGMMSG  MSGID(&MSGID) MSGF(QCPFMSG) MSGDTA(&MSGDTA) +
                            TOMSGQ(QSYSOPR)

             GOTO       CMDLBL(START)
ENDPGM
```

Figure A-1. Example of Sample Program to Receive Messages from QSYSMSG.

Appendix B. Sample Password Validation Program

The program in Figure B-1 is intended for use as a password validation program. If this CL program is compiled into library CLLIB with a program name of PASSWORD, command to set up the program as the password validation program is:

```
CHGSYSVAL SYSVAL(QPWDVLDPGM) VALUE('PASSWORD CLLIB').
```

```
 /**************************************************************/
 /* The password validation program contains 3 parameters.     */
 /* The new password (specified as &NEWPW here) and            */
 /* the old password (specified as &OLDPW here) are taken      */
 /* from the CHGPWD screen as input by the user.               */
 /* The return code (&RTNCODE) is a value set in this program  */
 /* and determines whether the new password is valid or not    */
 /* If a return code of 0 is passed back, the password is      */
 /* accepted.                                                  */
 /* If the return code is not 0, the password is rejected.     */
 /**************************************************************/
            PGM        PARM(&NEWPW &OLDPW &RTNCODE)
            DCL  VAR(&NEWPW) TYPE(*CHAR) LEN(10)  /* new password */
            DCL  VAR(&OLDPW) TYPE(*CHAR) LEN(10)  /* old password */
            DCL  VAR(&RTNCODE) TYPE(*CHAR) LEN(1) /* return code  */
 /**************************************************************/
 /* File PASSWORD in library COOPERS contains the              */
 /* passwords not to be accepted.                              */
 /**************************************************************/
            DCLF       FILE(COOPERS/PASSWORD)
 /**************************************************************/
 /* The return code is set to 0 as the default, meaning that   */
 /* the password is acceptable.                                */
 /**************************************************************/
            CHGVAR     VAR(&RTNCODE) VALUE('0')
READ:       RCVF      /* read the PASSWORD file */
            MONMSG     MSGID(CPF0864) EXEC(RETURN) /* quit at eof */
 /**************************************************************/
 /* The new password (&NEWPW) is compared to each password     */
 /* in the PASSWORD file (field name is PW).  If it matches,   */
 /* the return code is set to 1 (do not accept the password)   */
 /* and the program is ended (RETURN).  If it does not match,  */
 /* the next password in the PASSWORD file is read             */
 /* (GOTO READ).                                               */
 /**************************************************************/
            IF         COND(&NEWPW = &PW) THEN(DO)
            CHGVAR     VAR(&RTNCODE) VALUE('1')
               RETURN
            ENDDO
            GOTO READ
ENDPGM
```

Figure B-1. Sample Password Validation CL Program. The program reads a file of words that are considered inappropriate as passwords, and rejects a new password entered by the CHGPWD command if it matches one of the words in the file.

The file of passwords used in this program, COOPERS/PASSWORD, was created with the data description specification shown in Figure B-2 on page B-2

```
A*  THIS FILE CONTAINS INVALID PASSWORD VALUES
A          R PASSWORD
A            PW            10A         COLHDG('INVALID' 'PWD')
```

Figure B-2. Data Description Specifications. DDS for the password Validation CL Program.

Records can be added to the COOPERS/PASSWORD file through Data File Utility (DFU) or any other available method.

Note: Because of the sensitivity of this data, it is necessary to maintain a high level of security over the program and data file used. *PUBLIC authority should be set to *EXCLUDE for the data file, and set to *USE for the program.

WARNING!

A password validation program represents a security exposure. It receives unencrypted passwords.

Appendix C. Security Officer′s Password

What happens if the security officer forgets his password? Or if he is unavailable for some reason? Other users with *ALLOBJ cannot completely duplicate his functions, especially since (in most installations) the security officer is also the security administrator (*SECADM).

One approach is to have the security officer keep a written copy of his password locked in the company′s safe. This can work if the security officer *never* changes his password without changing the written record. (Working through a list of passwords is one way to accomplish the same thing).

There are various problems with this approach. Another approach is to have a defined user (named RESETSEC in this example) that is never used, except in one circumstance. If it is never used, there is no need to change passwords at intervals. The only circumstance in which this user profile is used is to reset the security officer′s password. The company president or a senior manager would keep the (unchanging) password for RESETSEC locked away.

The RESETSEC user profile would have an initial program that causes the security officer′s password to be reset to QSECOFR. That is, the RESETSEC user profile would have INLPGM(FIXIT).

The fixit program must be owned by the security officer and run with adopted authority. (Obviously this program must be installed while the security officer, with his password, is available. It cannot be installed after the problem arises). This program′s access should be very restricted, of course. The program could be very simple:

```
FIXIT
   PGM
     CHGUSRPRF USRPRF(QSECOFR)  PASSWORD(QSECOFR)
     SIGNOFF
   ENDPGM
```

The situation described here may seem amusing, but it could be very real in a good, secure installation. An auditor should insist on some defined recovery procedure for the described situation.

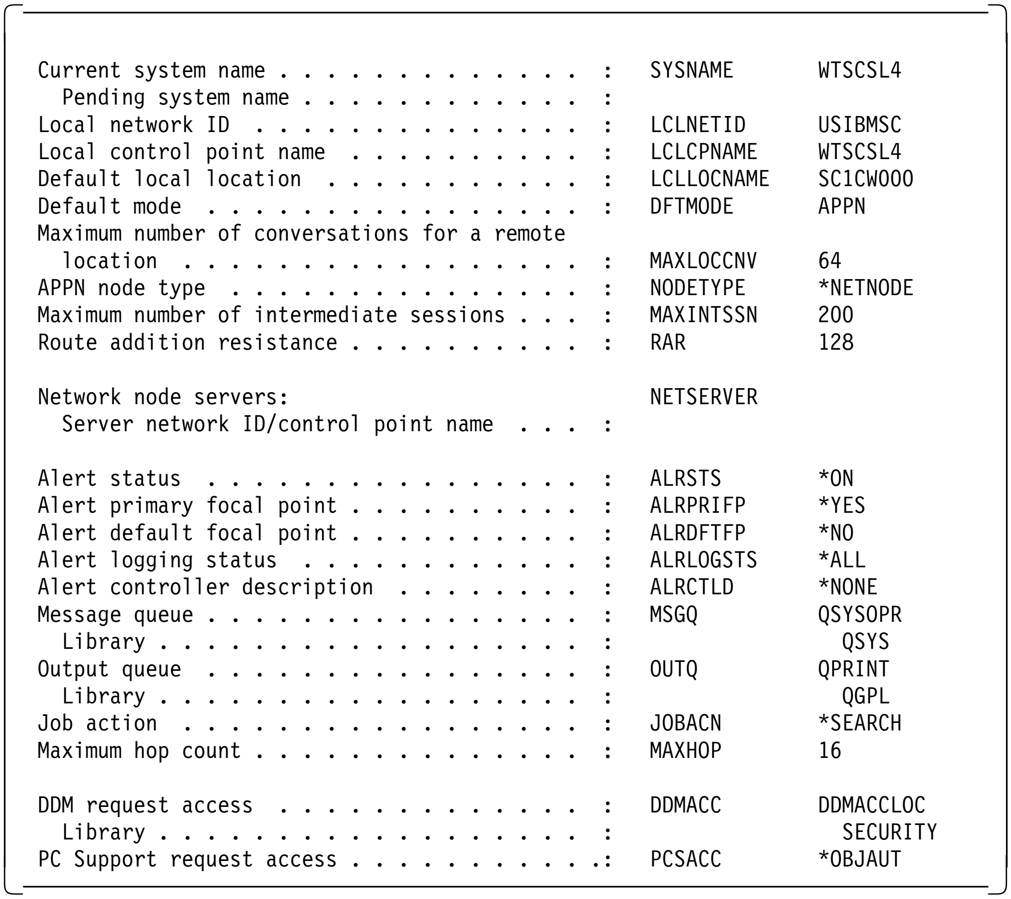
DST (dedicated service tools) also provides a method for recovery in this situation. However, DST requires usage skills that may not exist in all installations.

Appendix D. Program Used with DDMACC on Network Attributes

This program is called whenever a DDM request enters the system. It can be tailored to perform any form of validation your require.

```
/*********************************************************************/
/* This program checks the remote location from which DDM requests   */
/* come to this target System.  Only requests from remote location   */
/* SC1CW001 (SECURELOC *YES) are accepted.  In addition, the program */
/* checks to make sure that the user profile being used does not     */
/* start with ″Q″.  More sophisticated user profile checking could   */
/* be performed.                                                     */
/*********************************************************************/
/*                                                                   */
 PGM PARM(&RTNCODE &DATA)
             DCL        VAR(&DATA) TYPE(*CHAR) LEN(128) /* Input      +
                                                         Information */
             DCL        VAR(&RTNCODE) TYPE(*CHAR) LEN(1) /* Return    +
                                                                Code */
             DCL        VAR(&SRCLOC) TYPE(*CHAR) LEN(8) /* Source     +
                                                            Location */
             DCL        VAR(&USERID) TYPE(*CHAR) LEN(10) /* Target    +
                                                         User Profile*/
             DCL        VAR(&ZERO) TYPE(*CHAR) LEN(1) VALUE(’0’) /*   +
                                                   Reject DDM request*/
             DCL        VAR(&ONE)  TYPE(*CHAR) LEN(1) VALUE(’1’) /*   +
                                                  Accept DDM request */
/*********************************************************************/
             CHGVAR     VAR(&SRCLOC) VALUE(%SST(&DATA 76 10))
             IF         COND(&SRCLOC *NE ’SC1CW001’) THEN(DO)
             CHGVAR     VAR(&RTNCODE) VALUE(&ZERO)
                                  /* Reject DDM requests from Remote */
                                  /* Locations other than SC1CW001   */
             GOTO END
             ENDDO

             CHGVAR     VAR(&USERID) VALUE(%SST(&DATA 1 1))
             IF         COND(&USERID *EQ ’Q’) THEN(CHGVAR             +
                          VAR(&RTNCODE) VALUE(&ZERO))
                                    /* Reject DDM request for Target */
                                    /* User profiles starting with Q */

             ELSE       CMD(CHGVAR VAR(&RTNCODE) VALUE(&ONE))
                      /* If the remote location is SC1CW001 and the  */
                      /* Target User Profile specified does not start*/
                      /*  with Q, then accept DDM request            */
 END:        RETURN
             ENDPGM
```

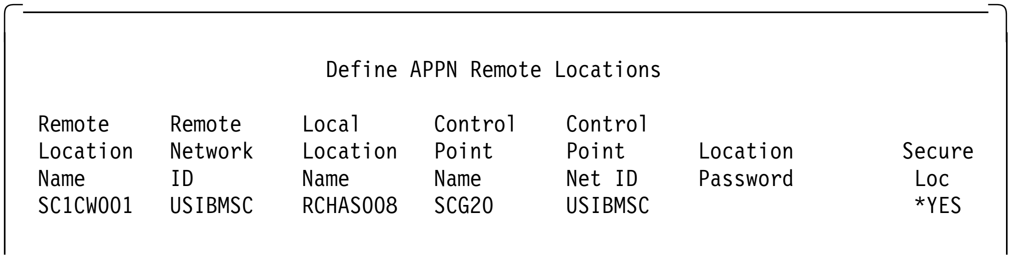
Figure D-1. Example Program for DDMACC Network Attribute. The program could be combined with the program used in Appendix E, “Example Exit Program for QRMTSIGN System Value” on page E-1 and called from both the DDMACC network attribute and the QRMTSIGN system value, for DSPT sessions. This would ensure that the same remote location could not be used for both DDM and DSPT sessions.



```
Current system name . . . . . . . . . . . . . :   SYSNAME       WTSCSL4
  Pending system name . . . . . . . . . . . . :
Local network ID  . . . . . . . . . . . . . . :   LCLNETID      USIBMSC
Local control point name  . . . . . . . . . . :   LCLCPNAME     WTSCSL4
Default local location  . . . . . . . . . . . :   LCLLOCNAME    SC1CW000
Default mode  . . . . . . . . . . . . . . . . :   DFTMODE       APPN
Maximum number of conversations for a remote
  location  . . . . . . . . . . . . . . . . . :   MAXLOCCNV     64
APPN node type  . . . . . . . . . . . . . . . :   NODETYPE      *NETNODE
Maximum number of intermediate sessions . . . :   MAXINTSSN     200
Route addition resistance . . . . . . . . . . :   RAR           128

Network node servers:                             NETSERVER
  Server network ID/control point name  . . . :

Alert status  . . . . . . . . . . . . . . . . :   ALRSTS        *ON
Alert primary focal point . . . . . . . . . . :   ALRPRIFP      *YES
Alert default focal point . . . . . . . . . . :   ALRDFTFP      *NO
Alert logging status  . . . . . . . . . . . . :   ALRLOGSTS     *ALL
Alert controller description  . . . . . . . . :   ALRCTLD       *NONE
Message queue . . . . . . . . . . . . . . . . :   MSGQ          QSYSOPR
  Library . . . . . . . . . . . . . . . . . . :                   QSYS
Output queue  . . . . . . . . . . . . . . . . :   OUTQ          QPRINT
  Library . . . . . . . . . . . . . . . . . . :                   QGPL
Job action  . . . . . . . . . . . . . . . . . :   JOBACN        *SEARCH
Maximum hop count . . . . . . . . . . . . . . :   MAXHOP        16

DDM request access  . . . . . . . . . . . . . :   DDMACC        DDMACCLOC
  Library . . . . . . . . . . . . . . . . . . :                   SECURITY
PC Support request access . . . . . . . . . . .:   PCSACC        *OBJAUT
```

.

Figure D-2. DDMACC Parameter on Network Attributes. When DDM requests arrive at the target AS/400, program DDMACCLOC is called. The program supplements normal AS/400 Object Authority checking.



```
                         Define APPN Remote Locations

Remote     Remote     Local      Control    Control
Location   Network    Location   Point      Point      Location          Secure
Name       ID         Name       Name       Net ID     Password           Loc
SC1CW001   USIBMSC    RCHAS008   SCG20      USIBMSC                       *YES
```

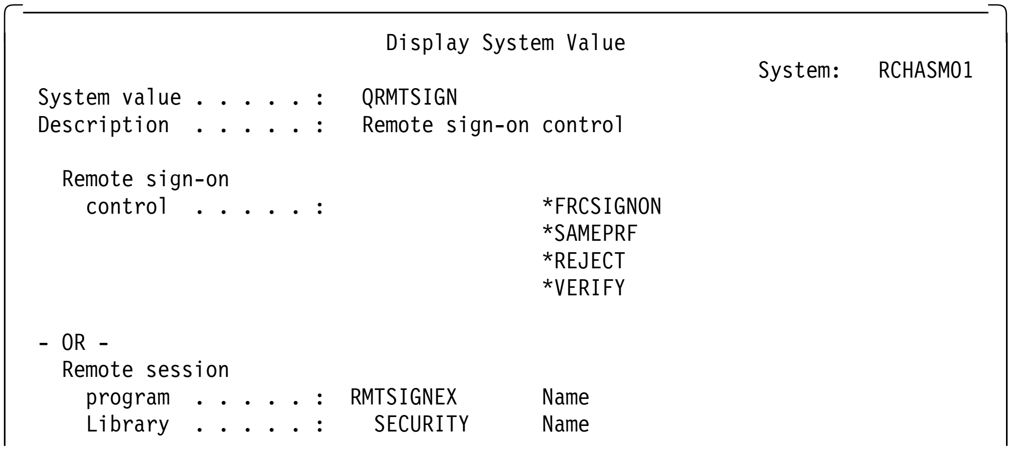
*Figure D-3. Remote Location List Entry for the DDM Remote Locations. This entry is necessary to identify the specific location from which the DDM request can be made. By allowing this to be SECURELOC(*YES), no password will be sent with the user profile - only the AVI. In this case a default user profile is not used.*

Appendix E. Example Exit Program for QRMTSIGN System Value

The following program can be used to validate which remote sessions will be allowed and which user profiles can be automatically signed on from which locations. It is activated by the QRMTSIGN system value.

```
/*********************************************************************/
/* This program checks the remote location from which DSPT requests  */
/* come to this target System.  Only requests from remote location   */
/* SC1CW000 (SECURELOC *NO) and by non ”Q” user profiles are         */
/* accepted.                                                         */
/*********************************************************************/

 PGM PARM(&DATA &RTNCODE)
             DCL        VAR(&DATA) TYPE(*CHAR) LEN(128)     /* Input +
                                                         Information */
             DCL        VAR(&RTNCODE) TYPE(*CHAR) LEN(8)   /* Return +
                                                                Code */
             DCL        VAR(&SRCLOC) TYPE(*CHAR) LEN(8)    /* Source +
                                                            Location */
             DCL        VAR(&USERID) TYPE(*CHAR) LEN(10)   /* Target +
                                                         User Profile*/
             DCL        VAR(&ZERO) TYPE(*CHAR) LEN(1) VALUE(’0’)
                                                 /* End DSPT session */
             DCL        VAR(&ONE)  TYPE(*CHAR) LEN(1) VALUE(’1’)
                                             /* Force Sign On screen */
             DCL        VAR(&TWO)  TYPE(*CHAR) LEN(1) VALUE(’2’)
                                          /* Allow Automatic Sign On */
             DCL        VAR(&WHENCALD) TYPE(*CHAR) LEN(1)     /* ’0’ +
                              ENDPASTHR, ’1’ force sign on, ’2’ auto +
                                                            sign-on  */
/*********************************************************************/
             CHGVAR     VAR(&WHENCALD) VALUE(%SST(&DATA 37 1))
             IF         COND(&WHENCALD *EQ ’0’) THEN(RETURN)
             CHGVAR     VAR(&SRCLOC) VALUE(%SST(&DATA 1 8))
             IF         COND(&SRCLOC *NE ’SC1CW000’) THEN(DO)
             CHGVAR     VAR(&RTNCODE) VALUE(&ZERO) /* Reject DSPT +
                          requests from Remote Locations  other than +
                          SC1CW000 */
             GOTO END
             ENDDO

             CHGVAR     VAR(&USERID) VALUE(%SST(&DATA 27 1))
             IF         COND(&USERID *EQ ’Q’) THEN(CHGVAR +
                          VAR(&RTNCODE) VALUE(&ONE))
                /* Force Sign on for User profiles starting with ’Q’ */
             ELSE       CMD(CHGVAR VAR(&RTNCODE) VALUE(&TWO))
                /* If the remote location is SC1CW000 and the Target */
                /* User Profile specified does not start with Q, then*/
                /* accept Automatic Sign on */
 END:        RETURN
             ENDPGM
```

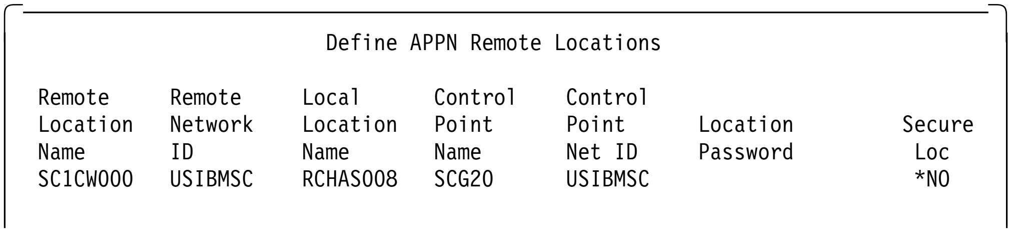
Figure E-1. Example Exit Program for QRMTSIGN System Value. The program could be combined with the program used in Appendix D, “Program Used with DDMACC on Network Attributes” on page D-1 and called from both the QRMTSIGN system value and the DDMACC network attribute for DDM functions. This would ensure that the same remote location could not be used for both DSPT and DDM sessions.



```
                          Display System Value
                                                       System:   RCHASM01
System value . . . . . :   QRMTSIGN
Description  . . . . . :   Remote sign-on control

  Remote sign-on
    control  . . . . . :                *FRCSIGNON
                                        *SAMEPRF
                                        *REJECT
                                        *VERIFY

- OR -
  Remote session
    program  . . . . . :  RMTSIGNEX     Name
    Library  . . . . . :    SECURITY    Name
```

Figure E-2. QRMTSIGN System Value for DSPT Exit Program. Display using DSPSYSVAL QRMTSIGN command. This shows the name of the exit program used for DSPT session requests, and the library containing the program.

Figure E-3 shows the Remote Location Configuration List entry that must exist for the RMTSIGNEX program. This combination ensures that only requests from remote location SC1CW000 will be accepted for a DSPT request to the target AS/400.



```
                     Define APPN Remote Locations

Remote     Remote     Local      Control    Control
Location   Network    Location   Point      Point                      Secure
Name       ID         Name       Name       Net ID     Location        Loc
SC1CW000   USIBMSC    RCHAS008   SCG20      USIBMSC    Password         *NO
```

*Figure E-3. Remote Location List Entry for the DSPT Remote Locations. This entry is necessary to identify the specific location from which the DSPT request can be made. By allowing this to be SECURELOC(*NO), the user will be forced to send a valid user profile and password.*

Appendix F. User Communications Application Programming Steps

Figure F-1 on page F-2 summarizes the steps involved in ICF programming. The example given is for an RPG program; however, the same process is used for other programming languages.

1. DDS source statements are created for a display file (ASDSP2 - if a display is presented during the application) and for the ICF file (ASICF1), as *members* in the file QDDSSRC. ASICF1 contains the formats for data to be sent across the communications link.
2. Source statements are used to create the Display and ICF files (with the CRTDSPF and CRTICFF commands). The files take the same names as the members from which they were created (display file ASDSP2 and ICF file ASICF1).
3. Program source statements are created as member(s) of RPG source file QRPGSRC.
4. Program source statements are compiled, using the CRTxxxPGM command, to create the program, having the same name as the source member, ASPGM3.
5. LIND, CTLD, and DEVDs are created for the communications link. The Remote Location Name of the DEVD is TAS400.
6. The Add ICF Device Entry command (ADDICFDEVE) is used to add a program device entry (ICF00) in the ICFF. ICF00 contains the Remote Location Name for the target application.
7. The Remote Location Name is also contained in the DEVD, which provides the connection to the physical communications link to the target site.

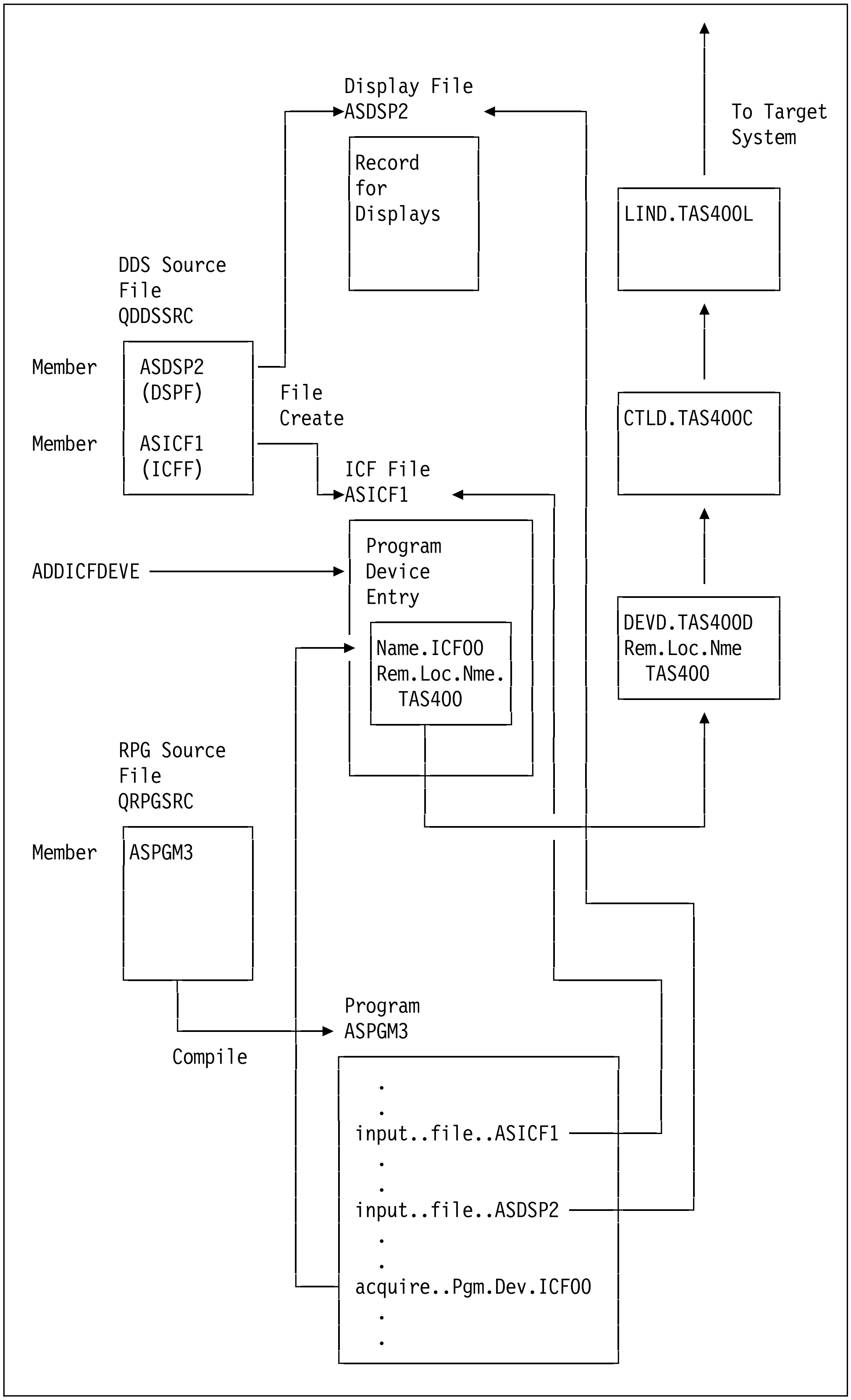


Figure F-1. ICF Programming. The diagram summarizes the steps involved in ICF programming See text for details.

Appendix G. Network User Profiles

This section is included to highlight considerations for user profiles when AS/400 is part of a network. Since no two networks are alike, it does not set out to provide all the answers for managing user profiles in a secure manner. Rather it discusses some of the possible issues that may be encountered and makes some suggestions, where appropriate.

The network administrator needs to make decisions about a range of questions on user profiles and passwords permitted in the network. For example:

- If a user is to work on multiple systems, should they have the same profile and/or the same password on all systems?
- Should the name of the user of any particular job be able to be determined, or can default profiles be used for network access?
- Should devices be varied off and user profiles be disabled after invalid signon attempts?
- How should security violations be reported?
- What profiles will be used to perform network support activities, such as problem diagnosis and change management?
- Should users have the same resource access rights on all systems?
- Should they be able to sign on to one AS/400 multiple times with the same profile?
- Who will use PC Support/400, DDM, SQL, and Office?
- Should there be a security officer at each distributed site, or one central security officer?
- Who will be responsible for maintaining user profiles?
- Will the system directories be shadowed or maintained manually?

All of these questions may be answered differently by different customers depending upon network size, mixture of system types, skill levels at distributed sites. The implications for their business will be different if security is not properly utilized or fully understood.

Ideally, user profiles should be unique for each user in a network. This means that if there is a user with a profile called WILSON on one system, that user profile should not be created on any other system unless the same person will use it. If this is not done, remote users may gain access to objects that they have no authority to, just because they have the same profile as someone who is authorized. Also, by keeping a user′s profile name consistent across a network, administration is simplified, and network usability is enhanced as users don′t need to remember multiple profile names.

Passwords need not be kept the same if locations are defined as secure. This means that if a password is illegally obtained, it cannot be used to gain access directly to other systems using Display Station Pass-Through (DSPT). On the other hand, if a location is defined as secure, applications such as Distributed Data Management (DDM) can be used to access remote files without providing a password. User-written APPC programs can prompt for the password to use on the remote system, otherwise just send the user ID with AVI.

When no user ID is sent, a default may be used on the remote system. This makes it impossible to determine which user is actually accessing files and issuing commands. This is a serious concern when DDM or user applications are being used.

Appendix H. Security Checklist

The following checklist can be used by auditors or system administrators to evaluate security setup on an AS/400 system. This checklist is broken down by function to give good principles for every system.

Before you start on this security checklist we encourage you to take a look at the *Basic Security Guide*. It has a number of forms that may be very useful in a security audit.

Physical Security

- The machine room should be water-proofed and fire-proofed. The door can be locked to control the entrance. Each entrance should be logged.

- UPS (Uninterruptable Power Supply) should be used to allow for a normal shutdown in case of a power outage.

- Physical access to the system console should be restricted.

- Recording confidential information such as user passwords on workstation/terminal record/play keys should be prohibited.

- Backup tapes and documentation should be protected from damage and theft.

- The key should be removed from AS/400 system panel and stored in a secure location. The keylock switch setting on the processor unit should not be in the manual position.

- The following questions should be asked:

  Are users of non-programmable terminals allowed to store information in the keyboard.

  Should the signon screen warn about unauthorized/illegal attempts to log on to the system.

System Values/Network Attributes

- The following questions should be asked:

  Who has authority to change system values, network attributes and work management? Are such changes documented and filed?

- Signon for users with *ALLOBJ or *SERVICE special authority should be limited to specific devices.

  System value QLMTSECOFR should be set to “1” to restrict users with *ALLOBJ or *SERVICE special authority to specific devices.

- System value QSECURITY should be set to 30 or higher to activate resource security.

- Security-related system values and network attributes should follow recommended guidelines.

  Use the following two commands, WRKSYSVAL (Work with System Value) and DSPNETA (Display Network Attributes), to list all security-related system values and network attributes:

  ```
  WRKSYSVAL SYSVAL(*SEC) OUTPUT(*PRINT)

  DSPNETA OUTPUT(*PRINT)
  ```

  Refer to the *AS/400 Security Reference* for information about the valid parameters.

- Decisions about system values and network attributes should be reviewed periodically, particularly when the system environment changes, such as the installation of new application or a communication network.

  If you activate QAUDJRN any changes to system values and network attributes can be logged. Refer to the *AS/400 Security Reference* about how to activate QAUDJRN.

User and Group Profiles

- Naming conventions for user profiles, group profiles and authorization lists should be followed.

- Each user should be assigned a unique profile. The system value QLMTDEVSSN should be set to “1.”

- Users should be limited to signing on at only one device at a time.

  System value QLMTDEVSSN should be set to “1” to limit users to one signed on device.

- Users should be able to change their own passwords. Allowing users to define their own passwords reduces the need for users to write down their passwords.

  Users should have access to the CHGPWD (Change Password) command or to the Change Password function from the Operational Assistant menu.

- Users that are limited to menus should have LMTCPB(*YES) specified to prevent override of initial program or initial menu at signon. This also restricts use of commands on system menus.

  See “Privileged Authorities” on page 14-23 for more information. But you should modify the query definition to list all user profiles with LMTCPB(*YES).

- Programmers should be restricted from production libraries.

  Use the DSPOBJAUT (Display Object Authority) command to determine the public and private authorities for production libraries and critical objects in the libraries. The administration of user profiles should be adequately organized. No user profiles should have large numbers of private authorities.

- There should be a routine on how to register a new user. Refer to “Register a New User” on page 14-4.

- Employees should be removed from the system immediately upon notification of transfer or termination.

  Regularly review the DSPAUTUSR (Display Authorized Users) list to make sure only active employees have access to the system. Refer to “Register a User Who Leaves” on page 14-4.

- User profiles should be checked to verify that they are not used as group profiles. See 3.4, “Group Profiles” on page 3-9.

- Owners of applications should verify the authorized users, including *PUBLIC access. The verification should be performed according to the company′s security policy.

- Management should regularly audit the users with special authorities, particularly *ALLOBJ special authority.

  See “Privileged Authorities” on page 14-23 for more information.

- Users that are limited to menus should have no menu option that allows entry of commands.

- The security officer profile or user with *ALLOBJ special authority should not be a group profile.

  If other profiles have the QSECOFR as a group profile, then these profiles should be controlled in a tight security environment. See “Privileged Authorities” on page 14-23 for more information.

- Group profiles **shall** have PASSWORD=*NONE.

- Group profiles should be identified with a naming convention.

  The naming convention GRPxxx for group profiles makes it apparent that multiple users are authorized when the group profile name is shown on a list of authorized users.

  Check the DSPAUTUSR (Display Authorized Users) command list:

  ```
  DSPAUTUSR SEQ(*GRPPRF) OUTPUT(*PRINT)
  ```

- Membership in a group profile should be changed when job responsibilities change.

  To verify group membership, use one of the these commands:

  ```
  DSPAUTUSR SEQ(*GRPPRF) OUTPUT(*PRINT)
  DSPUSRPRF USRPRF(group profile name) TYPE(*GRPMBR) OUTPUT(*PRINT)
  ```

Password Controls

- The passwords of IBM supplied profiles should be changed.

  If you are on release V2R3 or earlier verify that password of the following IBM-supplied user profiles have been changed: QSECOFR, QPGMR, QUSER, QSRV, QSRVBAS and QSYSOPR.

  If you are on V3R1 only QSECOFR needs to be changed. The rest of the Qxxxxxx profiles are shipped with password = *NONE.

  Refer to the *AS/400 Security Reference* for more information.

- Only a limited number of users should know the passwords for the Qxxxxx profiles above.

- The passwords for these Qxxxxxx profiles must be documented and kept in a safe place.

- Only authorized user should be able to change the passwords for the Qxxxxxx profiles.

- The IBM dedicated service tools (DST) passwords should be changed.

  Refer to the *AS/400 Security Reference* or 3.3.1, "Changing Passwords for Dedicated Service Tools (DST)" on page 3-8 for more information.

- Only a limited number of users should know the DST passwords.

- The DST passwords should be documented and kept in a safe place.

- There must be a rule for when the DST passwords are changed.

- Only a limited number of users should be authorized to change the DST passwords.

- The passwords of group profiles should be *NONE.

  Check the DSPAUTUSR (Display Authorized Users) command list:

  ```
  DSPAUTUSR SEQ(*GRPPRF) OUTPUT(*PRINT)
  ```

- Check when a user last changed the password.

  One way to do this is to copy all user profiles to an outfile. (QADSPUPB in QSYS can be used. It has all the necessary fields predefined). Run a query sorted on the date when the password was last changed.

- Password expiration active.

  System value QPWDEXPITV should be set to meet the organization′s security guidelines.

- Trivial passwords should be prevented by selecting QPWDxxxx system values.

  Use the WRKSYSVAL (Work with System Value) command to list all security-related system values beginning with QPWD: See 2.9, “System Values and Network Attributes Recommendations” on page 2-14 for more information.

- If a user profile has a password expiration interval that is different from the system value, it should meet or be better than the organization′s security guideline.

Applications and Ownership

- Applications in the production environment should be owned by a single user profile, not a group profile.

- Application ownership should correspond with the ″Plan to protect the business processes″.

- Documented routines should be followed when authorizing a user to an application.

- Authorization to applications should only be given to users who need it.

- Some objects within libraries may be secured better than others.

Programs

- Only specific libraries should contain source code.

- Request for changes to an application must be authorized and documented. The documentation must be kept.

- Changes must be done in a test environment.

- The programmer must document the changes made, and the documentation must be kept.

- It must be verified that the programs have been changed according to the programmer′s documentation.

- It must be verified that someone is responsible to check if changes to an application do not require a change in the backup routines, the restore strategy or the ″Plan to protect the business processes″.

- The ownership of the objects must be changed when they are transferred to the production environment.

- Programs in production environment should prevent use of DEBUG facilities to change variables.

- The source for programs should be captured when programs are moved into the production environment.

- The SECURE parameter in all override commands should be specified as *YES to prevent the file name being redirected to another file.

- Control the library list in applications to prevent a library that contains a similar program being added before the production libraries.

  You should specify a library name instead of default *LIBL in the source for programs.

- Programs that adopt authority must be checked for ownership, where they are used, and who has access to them.

- DFU and SQL should only be accessible by those who need them.

Authorization Control

- Owners of data should understand their obligation to authorize users on a need-to-know basis.

- Written forms should be signed by the application owners when a user is authorized to an application. Refer to 14.4.6, "Applications and Ownership" on page 14-5 for more information.

- Data should not be over-protected.

  System performance is improved when *PUBLIC authority is used for objects that do not justify protection. This also saves time during a backup of the system (SAVSYS or SAVSECDTA).

- Sensitive data should need public *EXCLUDE.

  Check the authority for user *PUBLIC for critical objects using DSPOBJAUT (Display Object Authority) command.

- Authorization should be defined at the library level where possible.

- The public authority to user profiles should be *EXCLUDE.

- Users are not allowed to sign on by pressing the Enter key on the Sign On display.

  Make sure no workstation entries in subsystem descriptions specify a job description that has a user profile name specified for the USER parameter.

- Authorization lists should be used to secure physical files. If the physical file has multiple members the use of an authorization list will provide better performance on save/restore than individual authorities.

- When the S36 environment is used authority holders must be checked.

- When the authorization to an object in the QSYS library is changed, it must be changed in all QSYSxxx libraries on the system, such as QSYS38 and QSYSVxRxMx.

- Job descriptions with *PUBLIC authority should specify USER(*RQD).

  To find out what job description are on the system, use

  ```
  DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*JOBD) DETAIL(*BASIC) OUTPUT(*PRINT)
  ```

  To check the USER parameter of a job description, use the DSPJOBD (Display Job Description) command.

- Job descriptions that specify a user profile name should have public authority *EXCLUDE, and should be authorized to specific users.

  To find out what job description are on the system, use

  ```
  DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*JOBD) DETAIL(*BASIC) OUTPUT(*PRINT)
  ```

  To check the authority to a job description, use the DSPOBJAUT (Display Object Authority) command.

Auditing Access

- Activate logging of security relevant events.

  System value QAUDLVL should be at least *AUTFAIL and *PGMFAIL. Regularly reviewing entries in the audit journal is the best method for detecting unauthorized attempts to access information.

- Entries in audit journal that report authorization failures should be reviewed for repeated offenders.

  Authorization failures cause AF type entries in the audit journal.

- Periodically review of changes to user profiles.

  Use the OUTFILE option on DSPUSRPRF to detect changes in user population and authority assignments.

- System value QMAXSIGN should limit number of access attempts to five or less. The QMAXSGNACN system value should be set at two or three.

- Message queue QSYSMSG should be created in library QSYS and monitored.

- The message CPF1116 shown when the user is about to exceed the retry limit for passwords should be changed to be the same as the invalid password message CPF1107. This change prevents the user from knowing that the next attempt will notify the security officer.

Communications

- It should be verified if users on other systems can log on to this AS/400. If so, which users and which systems?

- Which telecommunication lines are used for what purpose should be determined. Should they be online at IPL?

- Dial-in support should be protected by call-back procedures.

- Encryption should be used on sensitive data.

- Autoconfiguration should be prohibited once initial configuration has been done.

  System value QAUTOCFG should be set to “0” to turn off autoconfiguration.

- Remote signon should be controlled.

  The system value QRMTSIGN should be set to *FRCSIGNON or a pass-through validation program is used.

- Subsystems should prevent usage of default user. A user profile should be required to start a session.

- Access to data from other systems, including personal computers, should be controlled using the JOBACN, PCSACC, and DDMACC network attributes.

  Use the DSPNETA (Display Network Attributes) command to list all security-related network attributes:

  ```
  DSPNETA OUTPUT(*PRINT)
  ```

  See 2.9, “System Values and Network Attributes Recommendations” on page 2-14 for more information.

- Configuration lists must be checked.

- The use of DDM and ICF must be verified.

- APPC devices and remote controllers must be checked.

PC Support

- The folders should be checked for PC viruses periodically.

- Only those who need it should have access to the PC Support file transfer functions.

- PC users should not be allowed to store their passwords in the clear in a file in the PC.

OfficeVision

- Only a limited number of users should be able to execute commands from a document.

- Users who only works within OfficeVision may not need access to the Decision menu.

Appendix I. Security Tools in Library QUSRTOOL

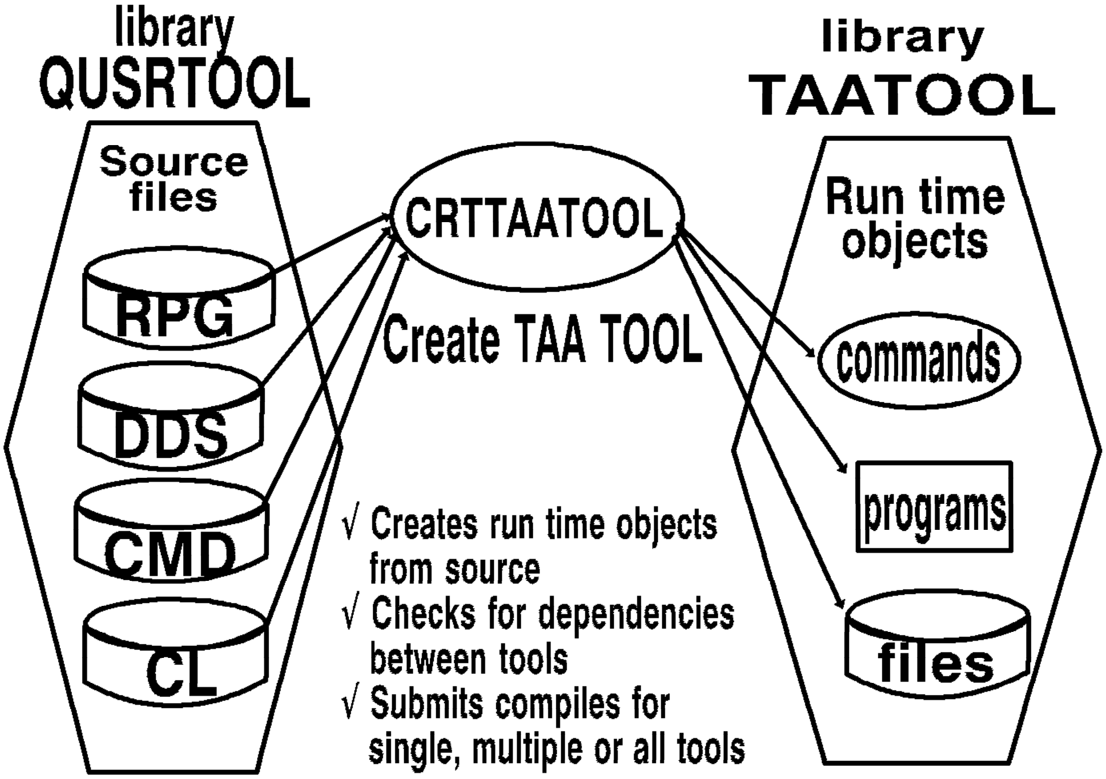
Library QUSRTOOL is an optional installable feature of OS/400. The library contains a number source files. The source files include useful tools for both the system auditor and security administrators. QUSRTOOL is a valuable resource that can be:

- Used unchanged to perform functions
- Modified for your installation needs
- Used as a learning tool for programmers

QUSRTOOL is delivered on an ″as is″ basis.

The detailed description for each tool is stored in the members of file QATTINFO in the QUSRTOOL library.

One of the first tools that should be created is CRTTAATOOL. The CRTTAATOOL command can then be used to create the remainder of the tools. (There are some tools that CRTTAATOOL does not create). Figure I-1 shows the process to create the tools from the source files in library QUSRTOOL. The CRTTAATOOL command creates run-time objects into library TAATOOL.



library
QUSRTOOL
Source files
RPG
DDS
CMD
CL
CRTTAATOOL
Create TAA TOOL
√ Creates run time objects from source
√ Checks for dependencies between tools
√ Submits compiles for single, multiple or all tools
library
TAATOOL
Run time objects
commands
programs
files

Figure I-1. Overview of QUSRTOOL Creation Process

Table I-1 on page I-2 lists the security related tools that are available in the QUSRTOOL library. The table indicates what security tools are focused towards system audit purposes and those focused towards management of system security.

Command Name	Command Title Description of Function	Audit	Sys Mgt
ACCSECLIB	Access secure library (Allow display but not change) It is desirable to keep the application programmers or auditors from changing anything in the production libraries. However, these users may have a valid need to be able to display anything and possibly copy data or create duplicate objects from the production library. **Note** This tool can be used to allow auditors view but NOT modify system data and object authorities. This eliminates the need to grant an auditor *ALLOBJ special authority.	X	X
CHGGRPPRF	Change group profile The Change Group Profile during a job command allows the user to switch to a different group profile during a job. This provides a form of multiple group profiles.		X
CHGLIBOWN	Change object ownership of a library and its objects The CHGLIBOWN command changes the owner of a library and its objects to another owner. If an object in the library is already owned by the new owner, no change occurs. An option exists to remove any authority of the old owner to the object. If the library contains programs that adopt the owner′s authority a change of ownership could cause the program to fail, or represent a security exposure. Don′t use CHGLIBOWN until you know what will be the impact.		X
CHGUSRPWD	Change user password (Send to second system) The Change User Password command is intended for those environments where a change to a user password should be replicated on one or more other systems. Instead of using the normal CHGUSRPRF command to change a password, the change would be made using CHGUSRPWD. This allows the password and document password to be captured, masked, sent to another system using SBMRMTCMD, unmasked and applied on the other system with the companion command CHGUSRPWD2.		X
CHKJOBDUSR	Check JOBDs for USER parameter The CHKJOBDUSR command allows you to find the job descriptions which are specified with the USER parameter containing a user profile name as opposed to *RQD. These job descriptions can be used to breach security unless they are properly authorized. On a system with the QSECURITY level set to 40 or 50, the system will check the user is authorized to use the profile on commands such as SBMJOB. However, certain functions, such as auto start jobs, require that the name exist in the JOBD and are not checked when execution occurs. At level 30 or below, the system does not check the SBMJOB user to see if the user is authorized to the name in a JOBD.	X	
CHKLIBOWN	Check for consistent ownership in a library The Check Library Owner command checks the objects in a library and prints job log messages for all objects that are not owned by a specific owner. This is intended for those cases where you are trying to ensure that all objects in a library are owned by the same user.	X	

Command Name	Command Title Description of Function	Audit	Sys Mgt
CHKLMTCPB	Check limited capability for *USER type profiles The Check limited capability command checks the user profiles on the system to determine the users specified as both USRCLS(*USER) and LMTCPB(*NO). An option also exists to force all *USER types to LMTCPB(*YES). CHKLMTCPB is designed to assist in ensuring proper security.	X	
CHKUSRAUT	Check user authority independent of program adopt In some environments it is desirable to operate under an adopted profile and yet still have a need to check individual authority. The CHKUSRAUT command will allow you to check an object′s authority independent of the program adopt function. The user group authority is always included in the check.	X	
CPYUSRPRF	Copy user profile (Send to second system) The Copy User Profile command is intended to allow a simple command method of duplicating a user profile based on an existing profile. The command would normally be followed by the system supplied CHGUSRPRF command or the EDTUSRPRF TAATOOL command.		X
DSPAUDLOG	Display audit log The DSPAUDLOG command displays the contents of the security audit journal (QAUDJRN). The output is always printed to a spooled file. The default will use DSPSPLF to display the printed output. An option can be used to select the level of detail to be presented. See 6.4.6, “Display Audit Log Command” on page 6-12 for more information on this command.	X	X
DSPPWD	Display password (captures new passwords and allows the display of user passwords.). The AS/400 operating system does not allow a display of a user′s password. The passwords are kept on the system in an encrypted form. These techniques are used to minimize security considerations involving an accidental exposure of a password or a break in attempt. The Display Password command includes an exit program for the system value QPWDVLDPGM. This program records a scrambled version of the user′s password in a file. This can be unscrambled and displayed, using the display password function. **NOTE** Use of this tool is *not* a good security practice. It is documented here so that auditors are aware of its existence. If a program name is specified in the QPWDVLDPGM system value, its function needs to be checked.		X
DSPSECRVW	Display security review The Display Security Review command is designed for the Security Officer or an auditor conducting a security review. It allows the output from the DSPUSRPRF command to be analyzed by a variety of methods.	X	

Command	**Command Title**	**Audit**	**Sys**
Name	**Description of Function**		**Mgt**
DSPUSRAUT	Display user authority The Display User Authority command lets you review authorizations by combining the individual object authorities, group profiles and authorization lists. The intent is to duplicate the type of checking performed by the system so that you can ask the following types of questions: • Who can update FILEA? • What can USERX do to the objects in LIBY? • What can *PUBLIC do to the objects in LIBZ? The authorization checking includes: • Object authorizations • Authorization list authorizations • Group profile authorizations • *ALLOBJ authority checking • Library authority (it is printed also) DSPUSRAUT does not consider: • The use of program adopt - USRPRF(*OWNER) • Authority holders • Dynamic switching of group profiles	X	X
EDTOBJAUT2	Edit object authority 2 (Send to second system) The Edit Object Authority 2 command is similar to the system supplied EDTOBJAUT command. The intent of EDTOBJAUT2 is to allow the GRT/RVK commands that are executed to be captured so they may be logged or sent to another system.		X
EDTUSRPRF	Edit user profile (Send to second system) The Edit User Profile command is intended for those environments where the CHGUSRPRF command should be captured so that the command may be sent to a log or another system.		X
ENAUSRPRF	Enable user profile The Enable User Profile command is intended for trusted personnel that do not have security officer authority to change a user profile from the disabled to the enabled state.		X
PRTSECVIL	Print security violations The Print Security Violations command uses the output file from the CVTQHST to print the security violations found in QHST.	X	
RSTFIL	Restore files to special library The Restore File command is intended to be used in an environment where the users may only restore files (such as data files) to the system. The files must be restored to the RSTOBJ library. The intent is to allow the RSTFIL command to be publicly authorized so that any user who is capable of accessing the proper devices can restore data. The data is always restored to the RSTOBJ library. The user must then copy the data to the appropriate library using normal system security. This allows the normal use of restore to occur in a secure manner. If restore commands are public, users who have *SAVSYS special authority can replace any existing object on the system. The RSTFIL command prevents this. When a library needs to be restored with different object types, a solution would be to temporarily authorize a user to the RSTLIB command or control the function so it occurs in a secure manner.		X

Command	Command Title	Audit	Sys
Name	Description of Function		Mgt
RTVSPCAUT	Retrieve special authority The Retrieve Special Authorities command retrieves each of the special authorities for a user profile. This differs from the RTVUSRPRF command, which retrieves a 100byte string.		X
SCRAMBLE	Scramble bytes within a field The SCRAMBLE command scrambles or unscrambles bytes in a field according to a scramble code field passed with the command. The SCRAMBLE command is useful when masking data. Scrambling means that byte 1 of the data becomes byte 3, byte 2 becomes byte 7 and so on. The scramble code field passed as a parameter allows the scrambling algorithm to vary. **Caution** The scrambling of data is not as secure as encrypting the data. Encryption is the preferred method to protect data. However, this tool is better than leaving sensitive data in the clear if you are unable to obtain an export license for the cryptographic hardware and software.		X

Table I-1. Security Tools in Library QUSRTOOL

Appendix J. Trusted Network Interpretation (TNI)

The Trusted Network Interpretation (TNI) is issued by the National Computer Security Center (NCSC) as part of computer security guidelines and describes the network system security guidelines for the following purposes:

1. To provide manufacturers with a security standard for sensitive applications
2. To provide a metric by which to evaluate a network system that processes sensitive information
3. To provide a basis of security requirements in acquisition specifications

The TNI consists of two parts: Part I and Part II. Part I of the TNI is based on the Trusted Computer System Evaluation Criteria (TCSEC), which is defined for stand alone systems, and interprets evaluation classes defined in TCSEC to fit them into a network system environment. Part II of the TNI describes additional security services which cannot be covered by an extensive interpretation of TCSEC.

Though the TNI can be a good reference for establishing and evaluating a secured system, it contains too much to explain and some parts are inapplicable to AS/400 users. Thus, in this section we briefly introduce TNI guidelines and limit discussion to the ideas that are useful for AS/400 users. If you need to know more about the TNI in detail, refer to *Trusted Network Interpretation of The Trusted Computer System Evaluation Criteria, Version 1*, July 1987, NCSC-TG-005, published by the National Computer Security Center.

Before discussing the TNI, some definitions of words used in the TNI to are listed help readers understand its concepts.

J.1 Definitions of TNI Terms

Accreditation: The managerial authorization and approval granted to an Automatic Data Processing (ADP) system or network to process sensitive data in an operational environment, made on the basis of a certification by designated technical personnel.

Automated Information System (AIS): An assembly of computer hardware, software, and/or firmware configured to collect, create, communicate, compute, disseminate, process, store, and/or control data or information.

Certification: The technical evaluation of a system′s security features, made as part of, and in support of, the approval/accreditation process, that establishes the extent to which a particular system′s design and implementation meet a set of specified security requirements.

Component: An individual physical unit that does not provide a complete set of end user services.

Dedicated Security Mode: The mode of operation in which the system is specifically and exclusively dedicated to and controlled for the processing of one particular type or classification of information, either for full-time operation or for a specific period of time.

Discretionary Access Control: A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The control is discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject. Adopted authority of AS/400, for example, meets this condition.

Interconnected Accredited AIS: A network consists of previously accredited AISs.

Mandatory Access Control: A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (that is, clearance) of subjects to access information of such sensitivity.

Multilevel Device: A device that is used in a manner that permits it to simultaneously process data of two or more security levels without risk of compromise. To accomplish this, sensitivity labels are normally stored on the same physical medium and in the same form (that is, machine readable or human-readable) as the data being processed.

Multilevel Secure: A class of system containing information with different sensitivities that simultaneously permits access by users with different security clearance and needs-to-know, but prevents users from obtaining access to information for which they lack authorization.

Multilevel Security Mode: The mode of operation that allows two or more classification levels of information to be processed simultaneously within the same system when some users are not cleared for all levels of information present.

NCSC-evaluation (National Computer Security Center): The process in which the NCSC determines whether a commercial off-the-shelf (COTS) product satisfies the TCSEC. It places a product in one of four divisions: D, C, B, or A.

Network System: The entire collection of hardware, firmware, and software necessary to provide a desired functionality.

Network Trusted Computing Base (NTCB): The totality of protection mechanisms within a network system - including hardware, firmware, and software - the combination of which is responsible for enforcing a security policy.

NTCB Partition: The totality of mechanisms within a single network subsystem for enforcing the network policy, as allocated to that subsystem. It is the part of the NTCB within a single network subsystem.

Object: A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video display, keyboard, clocks, printers, network nodes, and so on.

Single-Level Device: A device that is used to process data of a single security level at any one time. Since the device need not be trusted to separate data of different security levels, sensitivity labels do not have to be stored with the data being stored.

Subject: An activity entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state.

System High: The highest security level supported by a system at particular time or in a particular environment.

System High Security Mode: The mode of operation in which system hardware and software are only trusted to provide discretionary protection between users. In this mode, the entire system, to include all components electrically and/or physically connected, must operate with security measures commensurate with the highest classification and sensitivity of the information being processed and/or stored. All system users in this environment must possess clearances and authorization for all information contained in the system. All system output must be clearly marked with the highest classification and all system caveats until the information has been reviewed manually by an authorized individual to ensure appropriate classifications and that caveats have been affixed.

System/Subsystem: A collection of hardware, firmware, and software necessary configured to collect, create, communicate, compute, disseminate, process, store, and/or control data and information.

Trusted Computing Base (TCB): The totality of protection mechanisms within a computer system - including hardware, firmware, and software - the combination of which is responsible for enforcing a security policy. It creates a basic protection environment and provides additional user services required for a trusted computer system. The ability of a trusted computing base to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (for example, a user′s clearance) related to the security policy.

Trusted Path: A mechanism by which a person at terminal can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be initiated by untrusted software.

Trusted Software: The software portion of a Trusted Computing Base.

Unified Networks: A network which is accredited as a whole because its AIS subsystems are so specialized or dependent on other subsystems of the network for security support that individual accreditation of such subsystems is not possible or meaningful with respect to secure network operation.

J.2 Interpretation of TCSEC

The Trusted Computer System Evaluation Criteria (TCSEC) provides evaluation criteria to classify stand alone systems and the Trusted Network Interpretation (TNI) is intended to apply the criteria and classes introduced by TCSEC to network systems by extending the interpretation of TCSEC definitions. The classes, C1, C2, B1, B2, B3, and A1, are defined using the criteria shown in Table J-1 on page J-4. The table shows what criteria should be satisfied by each certification class and how the requirement for each criteria varies from class to class. For example, class C2 must satisfy the same requirements as class C1 for system integrity, security features user′s guide, test documentation, and design documentation, and additional requirements for discretionary access control, identification and authentication, system architecture, system testing, and trusted facility manual, and newly defined requirements for object reuse and audit.

Table J-1. Relationship of Evaluation Criteria and Classes

Criterion		**Class**					
		C1	**C2**	**B1**	**B2**	**B3**	**A1**
Security Policy	Discretionary Access Control	N	C,A	S	S	C,A	S
	Object Reuse		N	S	S	S	S
	Labels			N	C	S	S
	Label Integrity			N	S	S	S
	Exportation of Labeled Information			N	S	S	S
	Exportation to Multilevel Devices			N	S	S	S
	Exportation to Single-Level Devices			N	S	S	S
	Labeling Human-Readable Output			N	S	S	S
	Mandatory Access Control			N	C	S	S
	Subject Sensitivity Labels				N	S	S
	Device Labels				N	S	S
Accountability	Identification and Authentication	N	A	C	S	S	S
	Audit		N	C,A	A	A	S
	Trusted Path				N	C	S
Assurance	System Architecture	N	A	C	S	S	S
	System Integrity	N	S	S	S	S	S
	System Testing	N	A	N	C,A	C	C,A
	Design Specification and Verification			N	C,A	A	C,A
	Covert Channel Analysis				N	C	A
	Trusted Facility Management				N	A	S
	Configuration Management				N	S	C,A
	Trusted Recovery					N	S
	Trusted Distribution						N
Documentation	Security Features User′s Guide	N	S	S	S	S	S
	Trusted Facility Manual	N	A	A	A	A	S
	Test Documentation	N	S	S	A	S	A
	Design Documentation	N	S	A	C,A	A	C,A

Note:

Where, blank:No requirements, N:New requirements, A:Additional requirements, C:Change of requirements, S:Same requirements as previous level.

The TCSEC gives concise description for each class as follows:

- Class C1: Discretionary Security Protection

  The TCB of a class C1 system nominally satisfies the discretionary security requirements by providing separation of users and data. It incorporates some form of credible controls capable of enforcing access limitations on an individual basis, that is, ostensibly suitable for allowing users to be able to protect project or private information and to keep other users from accidentally reading or destroying their data. The class C1 environment is expected to be one of cooperating data at the same level(s) of sensitivity.

- Class C2: Controlled Access Protection

  System in this class enforce a more finely grained discretionary access control than C1 systems, making users individually accountable for their actions through login procedures, auditing of security-relevant events, and resource isolation.

- Class B1: Labeled Security Protection

  Class B1 systems require all the features required for class C2. In addition, an informational statement of the security policy model, data labeling, and mandatory access control over named subjects and objects must be present. The capability must exist for accurately labeling exported information. Any flaws identified by testing must be removed.

- Class B2: Structured Protection

  In class B2 system, the TCB is based on a clearly defined and documented formal security policy model that requires the discretionary and mandatory access control enforcement found in class B1 system be extended to all subjects and objects in the ADP (Automatic Data Processing) system. In addition, covert channels are addressed. The TCB must be carefully structured into protection-critical and non-protection-critical elements. The TCB interface is well-defined and the TCB design and implementation enable it to be subjected to more thorough testing and more complete review. Authentication mechanisms are strengthened, trusted facility management is provided in the form of support for system administrator and operator functions, and stringent configuration management controls are imposed. The system is relatively resistant to penetration.

- Class B3: Security Domains

  The class B3 TCB must satisfy the reference monitor requirements that it mediate all access of subjects to objects, be tamper proof, and be small enough to be subjected to analysis and tests. To this end, the TCB is structured to exclude code not essential to security policy enforcement, with significant system engineering during TCB design and implementation directed toward minimizing its complexity. A security administrator is supported, audit mechanisms are expanded to signal security-relevant events, and system recovery procedures are required. The system is highly resistant to penetration.

- Class A1: Verified Design

  Systems in class A1 are functionally equivalent to those in class B3 in that no additional architectural features or policy requirements are added. The distinguishing feature of system in this class is the analysis delivered formal design specification and verification techniques and the resulting high degree of assurance that the TCB is correctly implemented. This assurance is developed in nature, starting with a formal mode of the security policy and a formal top-level specification of design. In keeping with the extensive design and development analysis of the TCB required of systems in class A1, more stringent configuration management is required and procedures are established for security distributing the system to sites. A system security administrator is supported.

For details about requirement for each criteria, refer to *Trusted Network Interpretation of The Trusted Computer System Evaluation Criteria, Version 1*.

Note: The AS/400 does not support mandatory access control and labels requirements. Thus, it cannot be classified into higher than class C2.

These criteria are applied to a network system being accredited as a single system. As you can easily see, this approach cannot be applied to every case. The larger the network system becomes, the more difficult it becomes to apply this evaluation method. Therefore, another approach to evaluating a network system is also discussed in the TNI and we briefly describe that approach later in J.3, "Evaluation of Network" on page J-7.

J.2.1 Evaluation of Network Security Services

It is apparent that the approach to evaluating network security requires different viewpoint than that for the stand alone system. Unfortunately Part I of the TNI inadequately covers security issues from the viewpoint of the communication network. We therefore need additional evaluation guidelines for the network system to fill the gap. To that end, TNI Part II provides additional network security criteria which are disjoint from those defined in Part I. It focuses on risks which reside between end systems on the network. In a discussion of network security services, the TNI borrows heavily from the ISO Opens System Interconnection (OSI) standards. Though the discussion of security services is closely related to the OSI standard, the TNI additionally has its own concept for evaluating network security:

1. The TNI introduces the concept of strength of mechanism which allows us to evaluate relative strengths of different mechanisms.
2. It gives an evaluation indicating to what extent the security services work as it is intended.
3. The TNI considers the protection against denial of service threats.

On evaluation of security services in network systems TNI Part II focuses on three criteria: *functionality*, *strength of mechanism*, and *assurance*. Each criterion is defined in the TNI guideline as follows:

- Functionality refers to the objective and approach of a security service.
- Strength of mechanism refers to how well a specific approach may be expected to achieve its objectives.
- Assurance refers to a basis for believing that the functionality will be achieved.

With these criteria, nine network security services, that is, authentication, communications field integrity, non-repudiation, continuity of operations, protocol based protection, network management, data confidentiality, traffic confidentiality, and selective routing, should be evaluated.

The TNI Part II evaluation currently has quite loose and rough ratings compared with the Part I evaluation. Network systems are evaluated with the terms none, minimum, fair, and good. When the service is not offered by the manufacturer, it is rated as not offered. In some cases, services are rated just as none or present, if it is most meaningful. Table J-2 on page J-7 shows the evaluation structure for network security services.

Table J-2. Evaluation of Network Security Service

Network Security Service		Functionality	Strength	Assurance
Communication Integrity	Authentication	None, Present	None to Good	None to Good
	Communications Field Integrity	None to Good	None to Good	None to Good
	Non-repudiation	None, Present	None to Good	None to Good
Denial of Service	Continuity of Operations	None to Good	None to Good	None to Good
	Protocol Based Protection	None to Good	None to Good	None to Good
	Network Management	None to Good	None to Good	None to Good
Compromise Protection	Data Confidentiality	None, Present	Sensitivity level	None to Good
	Traffic Confidentiality	None, Present	Sensitivity level	None to Good
	Selective Routing	None, Present	None to Good	None to Good

J.3 Evaluation of Network

There are two viewpoints for evaluating a network system: a single trusted system view and an interconnected accredited Automated Information System (AIS) view. Each of them has its own advantage. Thus the viewpoint should be chosen based on which one is more suitable for the network system to apply.

J.3.1 Single Trusted System View

TNI Part I requires the network system to be accredited as a single entity (that is, a single trusted system) by a single accreditor. The approach for evaluating a network system with a single trusted system view is to divide the system into components, rate each component′s security characteristics, and finally evaluate the security class of the network system by examining the composition of the components.

The advantage of this approach is the rated components can be reused for evaluating a different network system without reevaluation. Each component need not support all the policies required by the TCSEC, and component types are defined for each component instead of rating it with TCSEC policies. Four relatively independent categories are selected from requirements for class A1 to build a concise set of component type:

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Audit
- Identification and Authentication

For convenience sake, they are denoted as M, D, A, and I respectively, and eleven component types are additionally derived from them: MD, MA, MI, DA, DI, IA, MDA, MDI, MIA, IAD, and MIAD. Each component falls into one of these component types base on the function it provides and each component type is evaluated with a class defined by the TCSEC as shown in Table J-3 on page J-8. Thus, we can evaluate components using classes defined by the TCSEC.

Note that the AS/400 cannot be an M component and a composed component which contains an M component because it does not support the requirements for mandatory access control.

Table J-3. Component Type Maximum and Minimum Class

Component Type	Minimum Class	Maximum Class
M	B1	A1
D	C1	C2+
I	C1	C2
A	C2	C2+
DI	C1	C2+
DA	C2	C2+
IA	C2	C2+
IAD	C2	C2+
MD	B1	A1
MA	B1	A1
MI	B1	A1
MDA	B1	A1
MDI	B1	A1
MIA	B1	A1
MIAD	B1	A1

Note: The class C2+ is positioned between C2 and B3.

Composition Rules and Composition Rating: Each component must satisfy some conditions required for each component type when it is combined with another component. Those conditions are called composition rules. The composition rules are designed to assure the functionality of the composed components. For details about the composition rule, refer to *Trusted Network Interpretation of The Trusted Computer System Evaluation Criteria, Version 1*, July 1987, NCSC-TG-005. Once its functionality has been assured, the composition is given a rating as shown in Table J-4.

Table J-4 (Page 1 of 2). Composed Components and Composition Rating

Composed Component Type	Evaluation Class of Component				Composition Rating
	D	I	A	M	
D	E(D)	-	-	-	LE(D)
I	-	E(I)	-	-	LE(I)
A	-	-	E(A)	-	LE(A)
M	-	-	-	E(M)	LE(M)
DI	E(D)	C1	-	-	C1
	E(D)	C2	-	-	E(D)

Table J-4 (Page 2 of 2). Composed Components and Composition Rating

Composed Component Type	Evaluation Class of Component				Composition Rating
	D	I	A	M	
DA	>=C2	-	E(A)	-	min(C2,E(A))
IA	-	C2	E(A)	-	E(A)
MD	C2	-	-	B1	B1
	C2	-	-	>B1	B2
	C2+	-	-	E(M)	E(M)
MI	-	C2	-	E(M)	E(M)
MA	-	-	C2	B1	B1
	-	-	C2	>B1	B2
	-	-	C2+	E(M)	E(M)
IAD	C2	C2	E(A)	-	C2
	C2+	C2	E(A)	-	E(A)
MDA	>=C2	-	C2	B1	B1
	>=C2	-	C2	>B1	B2
	C2+	-	C2+	E(M)	E(M)
MDI	C2	C2	-	B1	B1
	C2	C2	-	>B1	B2
	C2+	C2	-	E(M)	E(M)
MIA	-	C2	C2	B1	B1
	-	C2	C2	>B1	B2
	-	C2	C2+	E(M)	E(M)
MIAD	C2	C2	-	B1	B1
	C2	C2	-	>B1	B2
	C2+	C2	C2+	E(M)	E(M)

Note:

E(X) denotes evaluation class of X-Component.

LE(X) denotes the lowest evaluation class assigned to any X-Component within the composed component.

min(Y,Z) denotes either class Y or Z which is not greater than the other.

J.3.2 Interconnected Accredited AIS View

The interconnected accredited AIS view recognizes a network system as a collection of AISs which are independently created, managed, and accredited. This perception closely matches to our actual system environment in many cases. Though this approach does not provide a class evaluation of the network but just tells us if there are any security problems in the network system, it helps us assess risks of the network. In particular, when we plan to interconnect our system to the network built by another party, it can be a good alternative to a single trusted system view approach to assess risks. Because we usually do not know what the security policy enforcement by each component of the network is like in such situations, it is virtually impossible to evaluate the security level of the network system as a whole by using the approach suggested in the TNI Part

I. The accreditation range and the interconnection rule are the major concern points for assessing risks with this approach.

Accreditation Range

The accreditation range defines the range of data sensitivity levels and is determined based on the accreditor′s judgement on the ability of AIS to exchange information within a designated, or an acceptable, level of risk. If an operation mode of an AIS is system high or dedicated, its accreditation range must be a system high sensitivity level and the AIS must be single level by definition. The AS/400 belongs to this group if it uses security level 30 or higher because it only provides discretionary protection. A multilevel AIS can be assigned an accreditation range which covers all sensitivity levels processed by the AIS. Thus, the label of exported information is equal to the actual sensitivity level of the information.

AIS1
AIS2
S-C
TS
S
C

C2	Evaluation Class	B3
S	Accreditation Range	TS - C
System High (SH)	Operating Mode	Multi-Level Secure (MLS)

Figure J-1. Example of Accreditation Range

Figure J-1 shows examples of accreditation range in a single level and multilevel secure system. The subsystem AIS1, evaluated as class C2, contains confidential (C) and secret (S) information but is not trusted to separate them. Therefore, it is accredited to operate in system high (SH) mode and its accreditation range is the single level secret (S). Note that an AIS may contain information which security level is lower than its accredited range. All information sent from AIS1 must be labeled with the system high sensitivity label and it may be manually assigned a lower security level on the destination system after being received.

On the other hand, the subsystem AIS2 is a multilevel secure system which contains top secret (TS), secret (S), and confidential (C) information, and can be assigned an accreditation range equal to the entire set of levels. In this case, the label of information sent to other systems is equal to its actual security level on AIS2.

Interconnection Rule

Any AIS that is connected to other AIS must follow the interconnection rule. That is, each AIS or component on the network should be separately accredited to operate in an approved security mode of operation and for a specific accreditation range. The component is accredited to participate in the network at those levels and only those levels. In other words, the information exported at a given sensitivity level can be sent only to an AIS which accreditation range contains the same level or a higher level.

Figure J-2 shows how the interconnection rule applies. Subsystem AISX is assigned system high mode which is usually assigned to unevaluated or class C rated AISs. Because this system is not trusted to segregate sensitivity levels, it does not have explicit labels but an implicit label (in this example it is TS, or top secret). In contrast, subsystem AISY and AISZ are multilevel systems and are required to have explicit labels. In this example AISY has explicitly labeled confidential (C) and unclassified (U) information, and AISZ has explicitly labeled confidential (C), secret (S), and top secret (TS) information. Because of the interconnection rule, system AISX can import and export information from/to TS level on system AISZ, and can only import C and U information from system AISY. Thus the imported information on the system AISX has various actual levels with the implicit TS label. In this case, an authorized reviewer must manually downgrade it. Confidential (C) information can be exchanged between system AISY and AISZ. Unclassified (U) information on system AISY can be exported to system AISZ and it is relabeled as C when it is imported.

As previously mentioned, the AS/400 can operate only in, system high mode. When you connect your AS/400 with other systems, you have to decide what level of information you will allow to exchange. In other words, you have to decide the accreditation range of your AS/400 as an accreditor.

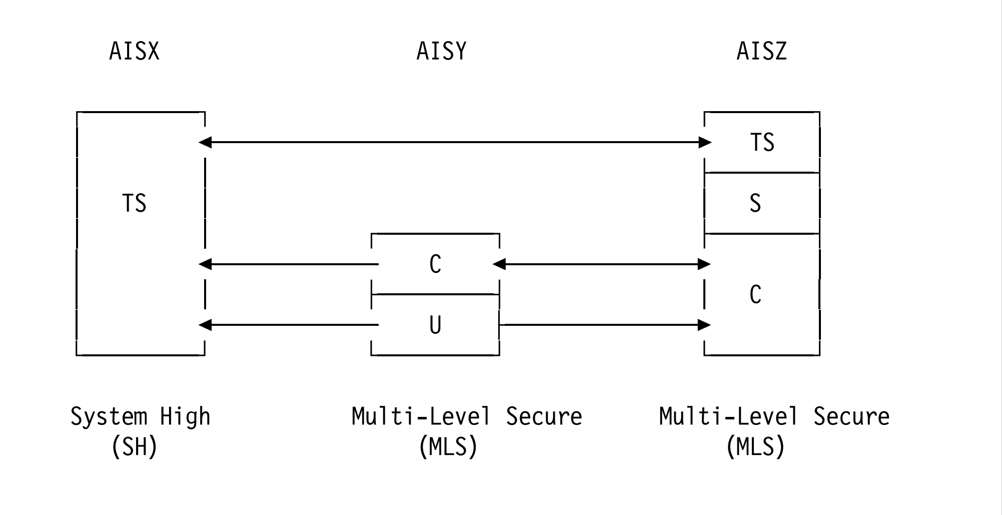


Figure J-2. Example of Interconnection Rule

Besides the accreditation range and interconnection rule, there are two security issues that must be considered when accredited AISs are interconnected. Those risks are known as propagation of local risk and the cascading problem.

Propagation of risks

Sometimes systems that do not have desirable class security exist in the network for some reasons (for example, operational needs). Connection of these systems or unevaluated systems to the network brings risks into the network. This situation is called propagation of risks. Therefore, it is necessary to take some security measures such as one-way connections, manual review of transmission, and data encryption when these systems are connected to the network.

Cascading problem

The cascading problem may happen when the systems that have different accreditation ranges and part of them are overlapped and are interconnected. Figure J-3 illustrates how the cascading problem occurs. An attacker first installs a malicious code, a so-called Trojan horse, in subsystem AIS1 (or AIS3) in order to leak top secret information to the secret level, then send it to subsystem AIS2 across the network. Finally the attacker installs a Trojan horse in subsystem AIS2 to leak the secret information which was originally a top secret to confidential. The path from top secret information in AIS1 (or AIS3) to confidential information in subsystem AIS2 is called cascading path.

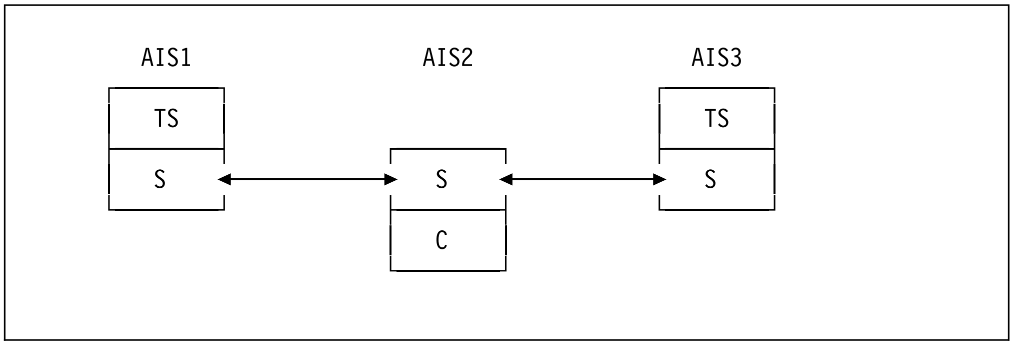


Figure J-3. Cascading Problem

There are two ways to solve the cascading problem:

- Use a more trusted system at appropriate nodes in the network.
- Use end-to-end data encryption.

The first solution does not perfectly prohibit the cascading problem in theory but it provides a barrier against the attacker which is more difficult to break. Thus we can reduce the risks of the network to the acceptable level. The second solution is effective when the communicating systems have same sensitivity levels, otherwise it is not applicable. For example, end-to-end encryption offers good protection for AIS1 and AIS3 in Figure J-3 but it is not true for AIS1 and AIS2 because they participate in a cascade and need to communicate.

To evaluate whether or not a cascading problem may exist, we can check the nesting condition. It is applicable only when all sensitivity levels are totally ordered; that is, if each of them can be uniquely sequenced in the order of its level. The nesting condition is satisfied if the accreditation ranges of each pair of AISs are disjoint or either of them includes the other. All possible pairs must be compared. If the nesting condition is met, there is no cascading problem. However, the failure of this test does not necessarily mean there is a cascading problem. It only indicates the possibility of the problem.

Index

ITSO Technical Bulletin Evaluation RED000

**An Implementation Guide for AS/400 Security and Auditing:
Including C2, Cryptography, Communications,
and PC Connectivity**

Publication No. GG24-4200-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

**Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

____

____

What other topics would you like to see in this Bulletin?

____

____

What other Technical Bulletins would you like to see published?

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

________________________ Name

________________________ Company or Organization

________________________ Phone No.

________________________ Address

____

____

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 977
3605 HIGHWAY 52 NORTH
ROCHESTER MN
USA 55901-7829

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

IBM ®

Printed in U.S.A.

GG24-4200-00