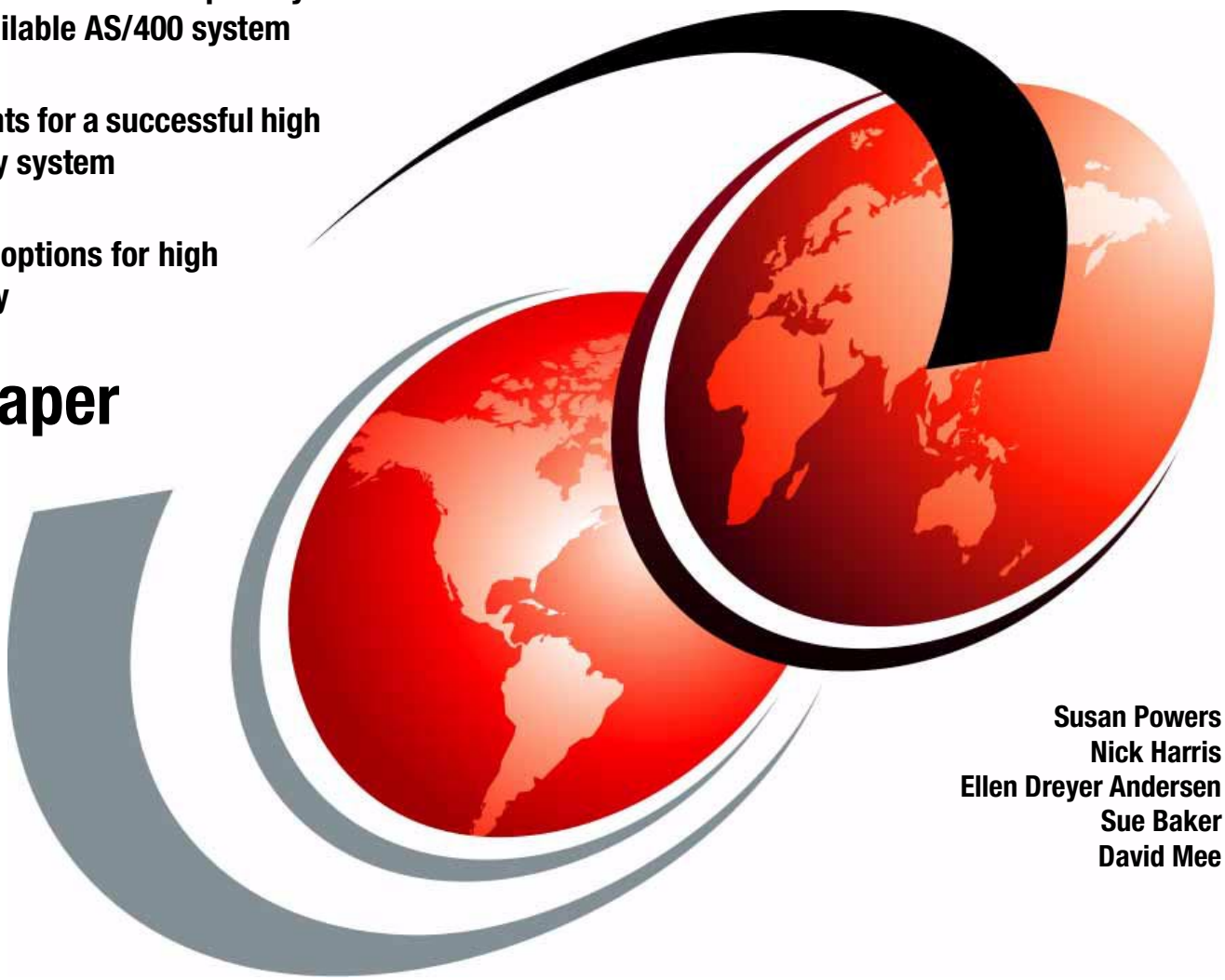
High Availability On the AS/400 System

A System Manager's Guide

- Tools and solutions to improve your highly available AS/400 system
- Components for a successful high availability system
- Hardware options for high availability

Redpaper



Susan Powers
Nick Harris
Ellen Dreyer Andersen
Sue Baker
David Mee

ibm.com/redbooks

Redbooks

International Technical Support Organization

High Availability On the AS/400 System: A System Manager's Guide

June 2001

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix H, “Special notices” on page 183.

First Edition (June 2001)

This edition applies to Version 4, Release Number 5 of OS/400 product number 5769-SS1.

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

Contents

Preface

Availability and disaster recovery represents a billion dollar industry in the United States alone. Professional associations and institutes, such as the Association of Contingency Planners, Business Resumption Planning Association, Contingency Planning and Recovery Institute, and their associated journals and magazines are devoted to keeping an information system (and, therefore, the business) available to both internal and external business users. The growth of e-business further emphasizes the need to maintain system availability.

Implementing a high availability solution is a complex task that requires diligent effort and a clear view of the objectives to be accomplished. The key to the process is planning and project management. This includes planning for an event, such as an outage, that may never occur, and project management with the discipline to dogmatically prepare, test, and perform for business resumption. Planning is paramount to the health of a highly available business.

This Redpaper is intended to help organize the tasks and simplify the decisions involved in planning and implementing a high availability solution. While some of the most relevent items are covered, this Redpaper cannot cover all cases because every situation is unique.

To assist IT managers with understanding the most important facts when planning to implement a high availability solution, detailed information is provided. This information can help business partners and IBMers to discuss high availability considerations with customers.

In addition, this Redpaper provides examples of highly available solutions, the hardware involved in AS/400 availability solutions, and OS/400 operating system options that add to the reliability of the system in an availability environment. Application software and how it affects an availability solution are also discussed.

Significant players in the solution are the business partners who provide the high availability middleware. In addion to discussing their products, a checklist is provided to help to establish a planning foundation.

Note: A service offering is available from IBM for examining and recommending availability improvements. Contact your IBM marketing representative for further information.

The team that wrote this Redpaper

This Redpaper was produced by contributions from a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Susan Powers is a Senior I/T Specialist at the International Technical Support Organization, Rochester Center. Prior to joining the ITSO in 1997, she was an AS/400 Technical Advocate in the IBM Support Center specializing in a variety of communications, performance, and work management assignments. Her IBM career began as a Program Support Representative and Systems Engineer in Des Moines, Iowa. She holds a degree in mathematics, with an emphasis in

education, from St. Mary's College of Notre Dame. She served as the project leader for this redbook.

Nick Harris is a Senior Systems Specialist for the AS/400 system in the International Technical Support Organization, Rochester Center. He specializes in server consolidation and the Integrated Netfinity Server. He writes and teaches IBM classes worldwide in areas of AS/400 system design, business intelligence, and database technology. He spent 11 years as a System Specialist in the United Kingdom AS/400 Business and has experience in S/36, S/38, and the AS/400 system. Nick served to outline the requirements and set much of the direction of this Redpaper.

Ellen Dreyer Andersen is an Certified IT Specialist in Denmark. She has 21 years of experience working with the AS/400 and System/3x platforms. Since 1994, Ellen has specialized in AS/400e Systems Management with a special emphasis on performance, ADSM/400, and high availabiity solutions.

Sue Baker is a Certified Consulting I/T Specialist working on the Advanced Technical Support team with IBM in Rochester. She has worked over 15 years with IBM mid-range system customers, in the industries of manufacturing, transportation, distribution, education, and telecommunications. She currently focuses on developing and implementing performance, capacity planning, and operations management techniques needed in the more complex multiple system and high availabiliy customer environment.

David Mee is a Strategic Accounts Project Manager in the Global Strategic Accounts Group of Vision Solutions. He specializes in application and database design, as well as integration and implementation of high availability solutions worldwide. He has over 15 years of experience in IBM Midrange systems, and holds a computer science degree with additional certificates from UCI and UCLA in RPG, Cobol, Pascal, C and Visual Basic programming languages. He writes and teaches classes on high availability, mirroring, and application resiliency for Vision Solutions.

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

Steve Finnes, Project Sponsor
Segment Manager, AS/400 Brand

Bob Gintowt, RAS Architecture
Availability/Recovery and Limits to Growth, IBM Rochester laboratory

Fred L. Grunewald
Vision Solutions, Inc.

Glenn Van Benschoten, Director Product Marketing
Lakeview Technology

Michael Warkentin, Senior Product Specialist
DataMirror

Comments welcome

Your comments are important to us!

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

- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Part 1. What is high availability?

Part 1 of this Redpaper discusses what high availability is. Levels of availability are discussed, as well as outage types, factors comprising an availability plan, and examples of high availability solutions.

Chapter 1. Background

Early systems were considered available when they were up and running. As the demands of business, communications, and customer service grew, systems had to be up and running through the normal working day (usually 8 to 10 hours). Failures during this working period were not acceptable. In availability terms, this was a 5 x 8 service (five days at 8 hours each day).

If a system was unavailable during this period, rapid recovery was necessary. Backups would be restored and the system and the database were inspected for integrity. This process could take days for larger databases.

These ocurrences eventually led to the definition of availability. In general, *availability* means the amount of service disruption that is acceptable to the end user.

This Redpaper provides insight into the challenges and choices a system manager may encounter when embarking on a project to make a business more highly available. This Redpaper does *not* provide a detailed technical setup of OS/400 or application products. This information is covered in other technical publications, for example the *AS/400 Software Installation* guide, SC41-5120.

1.1 When to consider a high availability solution

When considering if a high availability solution is right for you, ask yourself these questions:

- Will we benefit from using synchronized distributed databases?
- Do our users need access to the AS/400 system 24 hours a day, 365 days a year?
- Do our users operate in different time zones?
- Is there enough time for nightly backups, scheduled maintenance, or installing new releases?
- If our telephone sales application is not always up and running, will we lose our customers to the competition?
- Is there a single point of failure for any data center?
- Can we avoid the loss of data or access to the system in the event of a disaster or sabotage?
- When the production machine is overloaded, can we move some users to a different machine for read only jobs?

A high availability solution can benefit any of these situations.

1.1.1 What a high availability solution is

High (or continuous) availability systems usually include an alternate system or CPU that mirrors some of the activity of the production system, and a fast communications link. These systems also include replication or mirroring software and enough DASD to handle the volume of data for a reasonable recovery time as shown in Figure 1 on page 4.

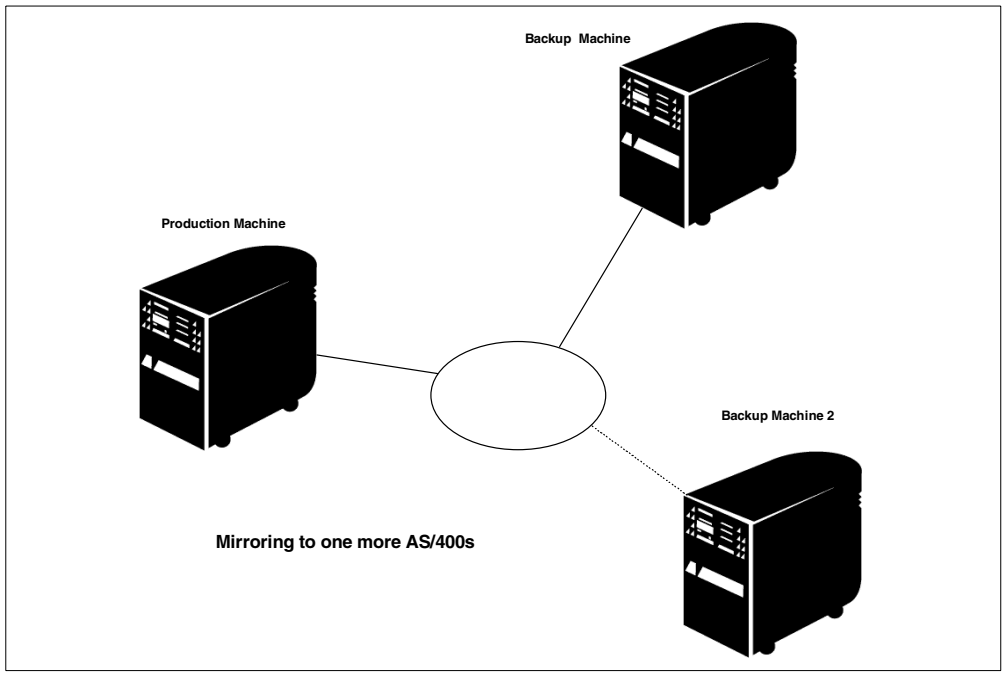


Figure 1. The basics of a high availability solution

Table 1 outlines some of the requirements of an HA solution. Tally these features to help simplify your investigation of high availability solutions.

Table 1. Requirements of a high availability solution

Features	Solution A	Solution B	Solution C	Data Propagator
24 x 7 availability				No
Eliminate downtime for backup and maintenance				No
Replication of database				Yes
Replication of other objects				No
Data replication to non-AS/400 systems				Yes
Handle unplanned outages				No
Automatically switch users to a target system				No
Workload distribution				Yes
Error recovery				Yes
Distribution to multiple AS/400 systems				Yes
Commitment control support				

Features	Solution A	Solution B	Solution C	Data Propagator
Sync checks				Yes
Filtering of mirrored objects				Yes (DB only)
Execute remote commands				No
OptiConnect support				Yes
Utilize Remote Journals				Yes

Figure 2 illustrates a business operating with high availability.

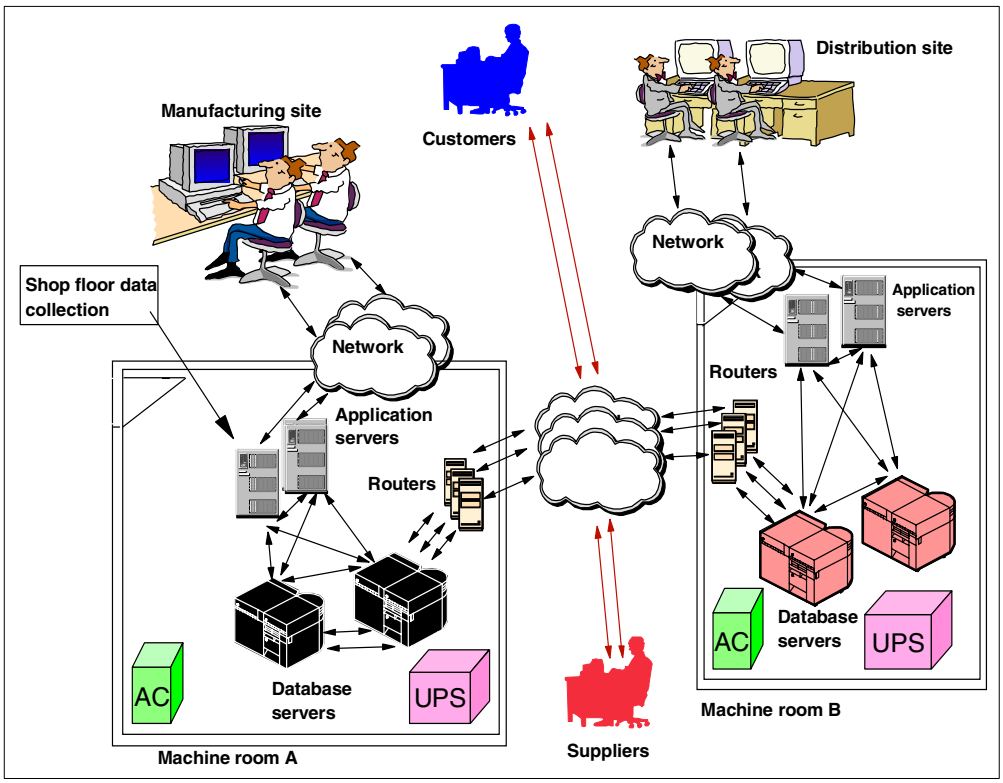


Figure 2. A highly available business

Note the redundancy of communications links, servers, the distributed work environment, and the use of backup power.

1.2 What high availability is

High availability is a much maligned phrase because it can have different meanings and is dependant on discussion variables. This Redpaper takes a holistic approach by discussing high availability as it applies to an organization, rather than an individual product or feature. This broadens the scope tremendously and prevents tunnel vision.

High availability in this context states that an organization, or part of an organization, can respond to a third-party request using the organization's *normal* business model. Normal is defined as including set levels of availability. Requests on the business can be anything, such as a sales call, an inventory inquiry, a credit check, or an invoice run.

High availability is achieved by having an alternative system that replicates the availability of the production system. These systems are connected by high-speed communications. High availability software is used to achieve the replication. Chapter 10, "High availability business partner solutions" on page 111, discusses these solutions for the AS/400e system.

1.2.1 Levels of availability

Information systems experience both planned and unplanned outages. Systems can be classified according to the degree to which they cope with different types of outages.

Your system can be as available as it is planned or designed to be. Orient your implementation choices toward your desired level of availability. These levels include:

- **High availability**: High availability relates to keeping an application available during planned business (service) hours. Systems provide high availability by delivering an acceptable or agreed upon level of service to the business during scheduled periods of operation.

  The system is protected in this high availability type of environment to recover from failures of major hardware components, such as a CPU, disks, and power supplies when an unplanned outage occurs. This involves redundancy of components to ensure that there is always an alternative available if something breaks. It also involves conducting thorough testing to ensure that any potential problems are detected before they can affect the production environment.

- **Continuous operations**: Continuous operations means that a system can provide service to its users at all times without outages (planned or otherwise). A system that has implemented continuous operations is capable of operating 24 hours a day, 365 days a year with no scheduled outage.

  This does not imply that the system is highly available. An application can run 24 hours a day, 7 days a week and still be available only 95% of the time because of unscheduled outages. When unscheduled outages occur, they are typically short in duration and recovery actions are unnecessary or minimal. The prerequisite for continuous operations is that few or no changes can be made to the system. In a normal production environment, this is a very unlikely scenario.

- **Continuous availability**: This type of availability is similar to continuous operations. Continuous availability is a combination of high availability and continuous operations. This means that the applications remain available across planned and unplanned system outages and must be implemented on system, application, and business levels.

  Continuous availability systems deliver an acceptable or agreed upon service 7 days a week, 24 hours a day. They add to availability provided by fault-tolerant systems by tolerating both planned and unplanned outages. With

continuous availability, you can avoid losing transactions. End users do not have to be aware that a failure or outage has occurred in the total environment.

In reality, when people say they need continuous availability, they usually mean that they want the application to be available at all times during the agreed service hours, regardless of problems with, or changes to, the underlying hardware or software. What makes this more stringent than high availability is that the service hours get longer and longer, to the point that there is no time left for making changes to any of the system components.

Note: The total environment consists of the computer, the network, workstations, applications, telephony, site facilities, and human resources. The levels of protection that a total environment offers depends on how many of these functions are wrapped into the integrated solution.

1.3 Determining your availability requirements

When most people are first asked how much availability they require, they often reply that they want continuous availability. However, the high cost of continuous availability often makes such requirements unrealistic. The question usually comes down to how much availability someone can afford.

There are not many applications that can justify the cost of 100% availability. The cost of availability increases dramatically as you get closer to 100%. Moving from 90% to 97%, for example, probably costs nothing more than better processes and practices, and very little in terms of additional hardware or software. Moving from 97% to 99.9% requires investing in the latest hardware technology, implementing very good processes and practices, and committing to staying current on software levels and maintenance.

At the highest extreme, 99.9% availability equates to 8.9 hours downtime a year. 99.998% equates to just 10 minutes unplanned downtime a year. Removing that last ten minutes of downtime is likely to be more costly than moving from 99.9% to 99.998%. What may be more beneficial, and less expensive, is to address planned outages. In an IBM study, planned outages accounted for over 90% of all outages. Of the planned ones, about 40% are for hardware, software, network, or application changes.

Appendix F, “Cost components of a business case” on page 169, helps you decide if the value of availability to the application justifies the expense.

1.4 Determining how high you need to go

It was previously mentioned that the former version of availability translated into a customer’s access to the business. A 9 a.m. to 5 p.m. time frame is a valid form of availability requirement. However, today the terms “24 x 7” or “24 x 365” are more commonly used. Even high availability is often substituted by the term “continuous availability”.

The term *The 9s* is also popular and means a 99% availability. On its face, this seems like a high requirement. However, if you analyze what this value means in business terms, it says that your process is available 361.35 days per year. In

other words, for 3.65 days, the process is not available. This equates to 87.6 hours, which is a huge and unacceptable amount of time for some businesses.

In recent press articles, up to $13,000 a minute has been quoted as a potential loss. This is a $68,328,000 a year lost potential. It is not difficult to justify a high availability solution when confronted by these sorts of numbers. However, it is one of the lessons to learn as well as an obstacle to block and tackle. The business can *potentially* lose this amount of money. When planning for a high availability solution, convincing the business to commit to even a fraction of this amount is difficult. However, use any recent unplanned or planned outages to estimate the cost.

It is recommended that you start with a departmental analysis. For example, how much would it cost the business if the salespeople could not accept orders for two hours due to a system failure?

To continue using *the 9s,* add .9% to your figure of 99%. This now equals 99.9% availability. Although this is obviously very close to 100%, it still equates to 8.76 hours. The AS/400e has been quoted as offering 99.94% availablility or 5.1 hours. This is according to a recent Gartner Group report, "AS/400e has the highest availability of any stand-alone, general business server: 99.94 percent" (*"Platform Availability Data: Can You Spare a Minute?*" Gartner Group, 10/98). This applies to the hardware and operating system and unplanned outages. It does not apply to application or scheduled outages.

1.5 Estimating the value of availability

The higher the level of availability, the higher the investment required. It is important to have a good understanding of the dollar value that IT systems provide to the business, as well as the costs to the business if these systems are not available. This exercise can be time consuming and difficult when you consider the number of variables that exist within the company. Some companies delay the analysis.

Once the value of the availability of your IT services is determined, you have an invaluable reference tool for establishing availability requirements, justifying appropriate investments in availability management solutions, and measuring returns on that investment.

The estimation process should:

- **Analyze by major application or by services provided**: The major cost of an outage is the cumulative total of not having the applications available to continue business.
- **Determine the value of system availability**: It is not easy to determine the cost of outages. The inaccessibility of each application or program has varying effects on the productivity of its users. Start with a reasonable estimation of what each critical application is worth to the business. Some applications are critical throughout major portions of the day, while others can be run any time or on demand.
- **Look at direct versus indirect costs**: *Direct costs* are the time and revenue lost directly because a system is down. *Indirect costs* are those incurred by another department or function as a result of an outage. For example, a

marketing department may absorb the cost of a manufacturing line being shut down because the system is unavailable. This is an indirect cost of the outage, but it is nonetheless a real cost to the company.

- **Consider tangible versus intangible costs**: *Tangible costs* are direct and indirect costs that can be measured in dollars and cents. *Intangible costs* are those for which cash never changes hands, such as lost opportunity, good will, market share, and so on.
- **Analyze fixed versus variable costs**: *Fixed costs* are direct, indirect, tangible, or intangible costs that result from a failure, regardless of the outage length. *Variable costs* are those that vary with the duration of the down time, but that are not necessarily directly proportional.

For more detailed calculations and methodology, refer *to So You Want to Estimate the Value of Availability*, GG22-9318.

1.6 iSeries factors for maximum availability

The IBM @server iSeries and AS/400e systems are renown for their availability due to a number of factors:

- **Design for availability**:

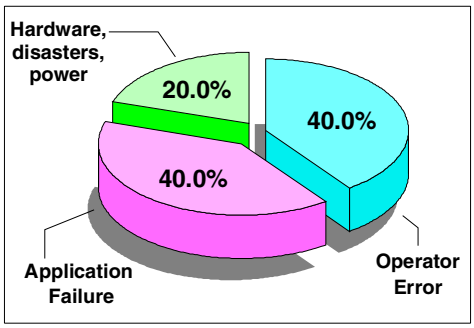


Figure 3. Unplanned outage factors

A single iSeries server delivers an average of 99.9+% availability. According to data collected by IBM over the last two years, AS/400 and iSeries owners have experienced an average of less than nine hours of unplanned down time per year. Figure 3 indicates two out of three factors of unplanned outages that can be affected by proper design. IBM delivers a very reliable server because the IBM Development team designs, creates, builds, tests, and services the iSeries and AS/400 systems as a single entity.

- **Effective system management process**:

As noted in Figure 4, 90 percent available translates to 36 days. Lack of attention to system management disciplines and processes affect the availability achieved. Availability solutions, such as clusters, are undermined when system management processes are lacking or nonexistent.

Availability Percentage		Total Outage Per Year
99.9999	=	32 seconds
99.999	=	5 minutes
99.99	=	53 minutes
99.9	=	8.8 hours
99	=	87 hours (3.6 days)
90	=	876 hours (36 days)

Figure 4. Unplanned outage factors

An effective system management strategy ties heavily into automation, such as an automatic archival of data, continuous system auditing, responding to security exposures, and monitoring error logs, backup and restores, and so on. A quote by Gartner Group puts this in perspective: *By the year 2003, "100% availability will remain elusive as user-controlled disciplines have an*

ever-greater relative impact on achieving availability" (Gartner Group, June Conference, Dallas, Tx., 1997).

An investment in system management disciplines, automation, and application recovery is necessary. Just a few additional hours of yearly downtime reduces availability from 99.99% availability to 99.9%.

- **Increase automation**:

  Increased availability means a reduction in the possibility of errors and recovery delays, and an increase in consistency. Human errors can create more down time than hardware failures or power outages. More effective automation through the use of automation software and tools can help offset an overburdened staff and allow them to attend to more unique and critical decisions and tasks. As availability requirements increase, investments in automation must also increase.

- **Exploit availability techniques and applications designed for availability**:

  Decrease unplanned outages and their effects by utilizing server availability options (for example, disk protection) and OS/400 functions, such as access path protection, file journaling, and user auxiliary storage pools (ASPs).

  Target a phased approach at increasing application resiliency (recoverability) and availability. As a general rule, an application on a non-clustered system is difficult to recover. A cluster solution cannot overcome a poor application design.

  Use applications that incorporate commitment control or other transaction methods, and use application recovery to decrease application recovery times and database integrity issues (incomplete transactions). You must use journaling and application recovery techniques to achieve high availability at a minimum. More sophisticated and highly available applications also use commitment control. Each technique is a building block foundation to all highly available environments.

- **Implement special solutions to meet your availability goals**:

  To reach your availability goals, special solutions, such as iSeries or AS/400 clusters with monitoring, automatic switchover, and recovery automation, are implemented to control both planned and unplanned outages. If you sidestep the issues described above, even sophisticated options like clusters may not provide the highest possible availability levels. Small outages, such as recovering or reentering transactions, add up.

1.7 Scheduled versus unscheduled outage

Many Information Technology departments have a good understanding of disaster recovery protection, which includes unscheduled outages. It can also involve anything from dual site installations to third-party vendors offering disaster recovery suites. Most of these installations provide protection from fires, floods, a tornado or an airplane crash on the site. The instance of the dangers affecting the site are very rare, but, if they do happen, they are catastrophic.

Scheduled outages can impact a business' financial well-being. Strong focus is warranted to plan for and minimize the time involved for scheduled outages. There has been little focus in this area, except for hardware and software vendors.

1.7.1 Scheduled outages

A scheduled outage is a form of planned business unavailability. Scheduled outages include a production line shutdown for maintenance, the installation of a new PABX, resurfacing the car park, or the entire sales team leaving town for a convention. All of these can influence the smooth operation of the business and, therefore, the customers.

In I/T terms, these outages may be due to a hardware upgrade, operating system maintenance or upgrade, an upgrade of an application system, network maintenance or improvements, a workstation maintenance or upgrade, or even a nightly backup.

The focus of outage discussion is shifting from unplanned outages (disasters, breakdowns, floods, and the like), to planned outages (primarily nightly backups, but also upgrades, OS/400 updates, application updates, and so forth). A growing solution is to consolidate individual systems onto large systems, rather than install distributed systems in departments or subsidiaries. This suggests that you may have users in different time zones, which can further minimize or eliminate the time available for routine operations, such as a backup. With a high availability solution and mirrored systems, customers can perform their daily backup on the backup system and let the users and work be done on the production system at the same time.

1.7.2 Unscheduled outages

Unscheduled outages include obscure occurrences. As mentioned earlier, these can include such things as fires, floods, storms, civil unrest, sabotage, and other assorted happenings. These outages are fairly well recognized and many organizations have disaster recovery plans in place to account for these types of occurrences. Testing these disaster plans should not be overlooked. We identify this in Appendix G, “End-to-end checklist” on page 175.

1.8 Comparison to planned preventive maintenance (PPM)

When researching this Redpaper, the authors found that there are only a few publications that relate to high availability for Information Infrastructure within organizations. They then looked laterally and found some good sources regarding practices and processes in the engineering aspect.

For many years, most manufacturing companies ran their businesses based on sound planned preventive maintenance programs. These programs cover the plant systems and services that support the manufactured product. Companies have long understood terms such as resiliency, availability, mean time to failure, and cost of failure.

To define the planned preventive maintenance schedule for a production line is a very complex operation. Some simple examples of the high level tasks that need to be performed include:

- **Documenting a business plan**: This includes business forecasts, current product forecasts, new product forecasts, and planned business outages (vacation, public holidays).

- **Documenting environmental issues**: This includes the frequency of power failures, fires, storms, floods. This also includes civil issues, such as unrest, strikes, layoffs, and morale.
- **Documenting the processes used**: This includes throughput, hours of operation, longevity of the product, and planned product changes.
- **Taking inventory of all parts involved in the process**: This includes purchase costs, purchase dates, history, quality, meantime to failure, replacement cycles, and part availability.
- **Documenting the maintenance and replacement process**
- **Estimating the cost for running the process and comparing it to the value of the product**
- **Estimating the affordability of a planned preventive maintenance program**
- **Documenting the resources required to manage the process**: This includes job specifications, critical skills, and external skills.

These tasks can be compared to the following tasks in the Information Management area. Imagine these refer to an application that supports one part of the business:

- **Documenting the business plan**: This includes business forecasts, current business forecasts that the application supports, business growth in this application, planned business outages (vacation, public holidays).
- **Documenting environmental issues**: This includes the frequency of power failures, fires, storms, floods. This also includes civil issues, such as unrest, strikes, layoffs, and morale.
- **Documenting the processes used**: This includes throughput, hours of operation, longevity of the application, and planned application changes.
- **Taking inventory of all parts involved in the application**: This includes purchase costs, purchase dates, history, quality, application failures, replacement cycles, hardware required to run the application, software maintenance schedules, software fix times, time required to implement ad hoc fixes, and developer availability.
- Documenting the maintenance and replacement process
- Estimating the cost for running the application and comparing it to the value of the business area
- Estimating the afford ability of making the application highly available program
- **Documenting the resources needed to manage the application**: This includes application specifications, job specifications, critical skills, and external skills.

This information is parsed into smaller sizes that can be applied to any business.

1.9 Other availability definition considerations

Designing a solution for high availability requires a practical knowledge of the business. The solution should fit the design of the business operations and structure. Does the organization have an accurate and approved business

process model? If it does, a major portion of the solution design is easy to plan. If there is no business process model, do *not* make assumptions at this stage. It is critical to get accurate information since this is the foundation of your plan.

Identify the systems and end users of each part of the business process model. Once identified, you have the names of people to relate, in practical terms, just how high is high availability.

Chapter 2. Developing an availability plan

For maximum availability, it is very important that planning is performed for the operating system, hardware, and database, as well as for applications and operational processes. As indicated in Figure 5, the level of availability is affected by products and use of technology. An unreliable system, poor systems management, security exposures, lack of automation, and applications that do not provide transaction recovery and restart capability weakens availability solutions.

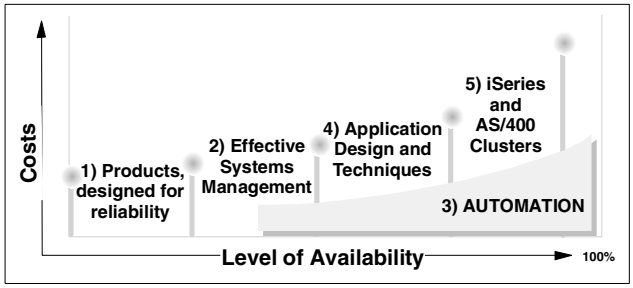


Figure 5. Essentials for maximum server availability

Achieving the highest possible level of availability is accomplished using clustering software, hardware solutions, and the planning and management of high availability products. An availability plan also needs to consider other factors that influence the end result, such as organizational and political issues. It is important to understand the challenges involved in each implementation phase to find the most appropriate tools and techniques for the customer environment.

The need for higher availability can be relaxed by determining what a business actually needs and what is possible using the available technology. In reality, most applications can withstand some planned outage, either for batch work, backups, and reorganization of files, or to affect application changes. In most cases, the application is expected to be available at the host, or perhaps on the network that is owned and controlled by the organization. However, continuous availability in the environment outside the control of the organization (that is, on the Internet) is not normally included in the business case.

This chapter explores the basic recommendations for planning for system availability. Start by looking at the needs of the business and the information necessary to support it. These actions are separated into the following areas:

- Reviewing the business plan
- Understanding the human resources issues
- Dealing with third party contracts

2.1 The business plan

This section gives the reader an insight into the information that is gathered from the business. Some of this is found in the business plan and some from investigating individual departments.

Why do you need all this information? To make the business and application highly available, you must take a wider view than just maintaining access to the information (system). It is not enough to have the most highly available system if the telephone system fails and no one can contact your company.

When you gather this information or interview staff within the company, you may be questioned as to why you need this information. Some co-workers may deny access to the information itself while some co-workers may roadblock your access to the people with the information. In cases such as these, your executive sponsor plays a very important role. The roadblocks you find can be easily avoided with the sponsor's help.

2.1.1 Project scope and goal definition

The business case, the driving reason for the high availability project, can be used as the starting point for defining the goals and understanding the scope of the project. Requirements and goals should be gathered from any department or party affected by system management. The project team must prioritize the requirements according to their urgency, importance, and dependencies, and to be able to define the short and long term goals.

Success criteria and completion criteria should be defined appropriately. Quantifiable results are essential. Even if it is an intangible benefit, such as customer satisfaction, you need to provide concrete methods to measure them.

To verify the requirements and goals, several starting points can be used:

- Look for established solutions for special requirements.
- Find out whether it is possible to develop a solution in a reasonable amount of time and with a reasonable amount of resource.
- Visit reference sites.
- Discuss the requirements with experienced consulting teams.

After this work is done, the first draft of the project definition is ready. Show it to the executive steering committee to get feedback as to whether it reflects their vision.

The results from this meeting are the central theme for elaborating on the detailed project definition. Input for this step includes:

- Requirements
- All information
- Type of politics
- Vision

The result is a clear and unambiguous definition of the scope and the goal definition.

2.2 Human resources

Producing an availability plan is not a single-threaded process, and it can *not* be built on one person's view. It is critical to build a team. The team should represent the needs of those they represent. Members must understand the department problems, be a voice for their challenges, be able to communicate their needs, and support an action plan.

A good project group is critical to this type of planning project. The activities of the business are diverse, represented by members bringing a diverse set of needs and views regarding both the problems and solutions,

There will be some very tough times, long hours, and unease as the project progresses. Determination, empathy, and leadership are valued skills for the team makeup.

Take a look at existing resources with an emphasis on your existing staff. In many cases, the skills required can be found within your staff. In other cases, it makes more sense to look outside the staff for the required talents. The final makeup of the team may represent both personnel inside and outside the company (or the department, if the study is of a more-focused nature).

2.2.1 Project organization

The organization of the project team can be categorized in two parts: the members of the team, and the other organization parties involved. Many different skills are required on the team. Criteria for the suitability of each team member includes their skills as well as their availability and personal characteristics, such as how they work on a team, negotiation skills, and knowledge of the technical solution, as well as the workings of the company. A suitable project manager should be selected from within or outside the company. The project manager is the “face” of the project and is responsible for the total health of the project, ensuring that the objectives and goals are met in the highest quality fashion within the specified time schedule and budget.

When the team is built, a lot of communication is necessary. Regular status meetings should be held to communicate and assess the actual status of the project, checkpoint the progress, make short-term decisions, and assess remaining activities. The sponsoring executives should be informed of the status in regular meetings.

Consider providing an office workspace for the project team, and supply it with applicable reference material for the subject area and company. Sometimes management chooses a project manager from outside the company. Sometimes an organization expert joins the project team to provide the project manager with knowledge about the internal structures of the company.

2.3 Communication and sponsorship

The support of the company’s management to the project is critical to its success. The ultimate responsibility for the project should lay with an executive steering committee, or an executive serving as a contact between the team and executives. The steering committee represents the management involvement and sponsorship. It can be used to solve complications with other parts of the organization, communicate expectations to affected company parties, and provide decision making power.

The vision of the project scope and objectives should be made clear to everyone involved. This vision must be broadly communicated through the entire organization so that everyone can be aware of the consequences the project has for them. Reactions, especially negative, should be taken seriously and should be used to reach a clearer project vision. Pay special attention to communication because this can prompt people to bring their ideas into the project.

Non-technical aspects should be emphasized in the project plan so that non-technical people involved can better adopt the project as their own. If it is appropriate, consider having a customer or supplier representative on the team.

2.4 Service level agreements

Service level agreements are contracts with business departments within your company and, in many cases, businesses with whom you have an external contractual relationship. If your network resources are not up and running, you can *not* keep these commitments.

Within your company, service requirements probably differ by department. For example, your accounting department may be able to tolerate down time of the accounting applications for up to one hour before it starts negatively impacting department operation. Likewise, marketing may agree to a calendar three-hour down time, but it can allow only an hour of down time on a customer-reference database.

You make service level agreements to track your IT organization's performance against business requirements. Based on these agreements, set priorities and allocate limited IT resources. By linking your IT department to your service desk, you can manage complex relationships among user problems, corporate assets, IT changes, and network events.

System management can provide a centralized view of your current asset inventory. This enables your analysts to correctly analyze and resolve problems. Help desk analysts work with administrators to plan and manage the effects of such IT changes as deploying and upgrading applications. System management involves tracking, logging, and escalating user interactions and requests.

2.5 Third party contracts

Service level agreements outside your company can mean a loss of business to a competitor if you can't meet your commitments. This can have serious consequences for your company.

Most organizations have contracts with external suppliers. However, the third parties are not all typically under the control of one department. As a Systems Manager implementing a high availability solution, you interface with many different aspects of your organization to gather the required information and possibly change the contracts to meet your new needs.

Contractors or resources utilized from outside the company represents programmers, technicians, operators, or a variety of consultants.

2.5.1 Application providers

The AS/400 system gets its name from Application System. The majority of AS/400 customers have one or many applications installed on their AS/400 system and the attached workstations.

When reviewing the application, establish how the provider supports your business and determine the required level of support. This can range from a 9 a.m. to 5 p.m. telephone support provider contract, to employing developers

skilled with the particular application. The more critical the application, the higher level of skills that are required to perform problem determination that enables the fix to be resolved in the shortest possible time.

The application provider should be able to define the rough release schedule for the application. This allows you to plan for application system updates. They should also be able to provide varying levels of fix support if the application fails in some way.

Develop an escalation process for critical failures. Think through and document what steps to follow to recover from a critical failure.

Note: This information applies to applications written outside the company. The same considerations apply for applications developed in-house.

2.5.2 Operating system provider

Operating system suppliers are similar to application providers. However, the opportunity to enhance the operating system code typically occurs more frequently than application providers code. Enhancements to the OS/400 operating system typically occur with an annual frequency. Updates occur more frequently on application software.

2.5.3 Hardware providers

Hardware provided by non-IBM distribution channels can include:

- CPUs
- Towers
- Tapes
- DASD
- IOPs
- Workstations
- Printers

Contracting with hardware providers is a relatively simple method to utilize resources outside the company. The contractor's reliability is normally well known and high. Maintenance organizations are highly skilled and can provide high levels of service, depending on the cost. At minimum, the hours of coverage for support should match your planned availability requirements.

Many large customers have arrangements for key supplies to be "warehoused" at the customer site. In other words, the supplies are owned by the maintainer but are stored at the customer site.

Be careful when ordering new hardware. Order only with a goal that ensures availability of spare parts early in the product life. Demand for a product can be so great that there are no spare parts available.

2.5.4 Peripheral equipment

The hardware components of a computing system go beyond the central processing unit and controllers. To reach end users, a computing system includes peripheral equipment, such as workstations, printers, routers, and modems.

Maintaining peripheral equipment can be difficult. You must judge the benefits of maintaining parts and components on-site for fast replacement, compared to a repair contract for the main equipment or a combination of both. Consider the longevity of the peripherals. For example, when printers, displays, modems and such are replaced, it is not uncommon to update the technology.

Sometimes, a replacement is made as an upgrade of functionality, either because the former model is withdrawn from sales and support, or the needs of the users require more function than the broken unit. It is not uncommon (perhaps even typical) for the replacement unit to provide equivalent or more function for less cost to the business.

At first glance, replacing technology may not seem to be a big problem. For example, consider the case of a printer connected to a personal computer that is used by others within the network.

Assume this departmental printer fails and is non-repairable. One solution is to simply replace the printer. However, after the new printer arrives and is connected, it needs a new driver loaded on the server operating system. The load of the new driver can raise a number of significant issues:

- Is an IPL required to be recognized by the system?
- Is a configuration required?
- Do the applications work with the new driver?
- What if the load causes the server to fail?

This circumstance can result in driver loads across the whole network that reduce the hours of productive business.

Therefore, even the smallest components of the total environment can have a major impact. It is advised that you plan for some redundancy, including when and how you carry out bulk replacements.

2.5.5 Facilities

Facilities include machine rooms, power supplies, physical security, heating and air-conditioning, office space, ergonomics, and fire and smoke prevention systems to name but a few. The facilities that complete the total environment are as complex as the applications and computer hardware.

2.5.5.1 Site services

It is important to work alongside the site facilities personnel and to understand contracts and service levels. For example, turning off the air conditioning for maintenance can potentially have a disastrous effect on the systems in the machine room. The contract with facilities should include spare air-conditioners or the placement of temporary mobile conditioners.

2.5.5.2 Machine rooms

A good system manager has a well documented understanding of the machine rooms and the equipment within. The same redundancy requirements should be placed on critical services just as there are on computer systems.

2.5.5.3 UPS

The use of Uninterruptible Power Supplies (UPS) has grown tremendously over the past ten years. The cost of a small UPS has reduced and they are now very affordable.

When considering a UPS, keep in mind these three broad areas:

- **Machine Room**: The machine room is a prime candidate for a UPS because it often requires continuous power. The ultimate solution is to generate your own power. Ideally, the national power supply serves only as a standby, with limited battery backup. Some customers do have this arrangement, but it is an expensive solution. There are simpler solutions that provide nearly the same level of service.

  Consider switching to a generated power environment. When the power fails, the standby generator starts. Unfortunately, there is a time lag before the system comes on line. Therefore, an interim battery backup is needed to support the systems while the generated power comes online. To register that the power has failed and then switch between battery, generator, and back to normal power requires a complicated and expensive switch. It may also require links to the systems to warn operators of the power failure.

  Another area that is often overlooked is the provision over power supplies for other equipment in the machine room, for example, consoles, printers, and air-conditioning. It is not enough to have a UPS for the system if there is no access to the console. In an extreme condition, you may be unable to shut the system down before the battery power fails if there is no UPS support for the console.

- **The site**: When considering the site, you must look at all areas, for example:

  – If you have a disaster recovery service, is there space to park the recovery vehicle in the car park close to the building to attach network cabling?

  – In some buildings, access to the machine room is a problem and systems must be moved through windows via a crane because the lifts are too small or cannot take the weight.

  – Are there high availability facilities in the general office space, such as an emergency telephone with direct lines in case the PABX fails. This allows the business to continue even if the desk phones fail.

  When developing the availability plan, document these restrictions and plan for circumventions.

- **The workstation**: Key users may need backup power to their workstation if there is no full standby generation.

  Looking at workstations from an ergonomic view, you may find potential issues that can result in long term unavailability of the human resources. An example is repetitive strain injury. This can severely impact your critical human resources in a company and cost a significant amount in litigation.

  It is worth investing time and money to solve these problems when planning for availability.

2.6 Verifying the implementation

Verifying (by testing) the proposed high availability solution before putting it into production cannot be over-emphasized. Some of the activities involved in testing the implementation are explained in the following list (several areas of verifying, or testing, are involved):

- **Build a prototype**: A prototype is a simulation of a live production. Develop a prototype to test the quality of the high availability solution. Make sure it can be easily reproduced. This lowers the risk of disturbing production during installation rollout.
- **Regression tests**: Ensure that the replicated solution works in the same way as the prototype by performing a number of regression tests in the new environment. These same tests should be used in the production environment to ensure the quality of the final high availability implementation.
- **Disaster recovery test**: Develop routines to ensure that a disaster recovery is smooth and as fast as possible. This requires a real crash test with detailed documentation of the steps required to get the system up and running, including how long it takes and where to find backup media and other required material.
- **Volume test**: Few companies have the resources to build a test environment large enough to do realistic tests with production-like volumes. Some recovery centers and business partners offer the use of their environments for performing tests in larger volumes. It is an important step to help ensure that the system behaves as expected during the actual production.

2.6.1 Documenting the results

To retest the systems after a problem has been fixed, it is important to have every test situation well documented. The documentation should include:

- Hardware requirements
- Software requirements
- HAV requirements
- A test case category
- Tools required to perform the test
- A list of steps to execute the test case
- Results of the test (pass or fail)
- The name of the individual executing the test case
- The date on which the test case was executed
- Notes taken while the case is executed
- Anticipated results for each step of the test case
- Actual results
- Comments on general notes for each test case.
- A list of any problem records opened in the event that the test was not successful

2.7 Rollout

The testing sequence ends with the confirmed rollout of the solution. The rollout is another milestone in the overall project. For the first time, the production environment is actually affected. A well designed rollout strategy is crucial. Many things can impact the success of the rollout. Carefully check all prerequisites.

In a typical rollout, a great number of people are affected. Therefore, communication and training is important. For the rollout into production volumes, sufficient support is required. A minor incident can endanger the production rollout.

The basic infrastructure of the company influences the rollout process. The situation differs depending on the industry area.

The risk of problems increases with the number of external factors and the complexity of the system. The ideal case is a monogamous environment with all equipment owned by the company, including a high-performance network. All success factors can then be planned and controlled within individual departments.

The main issue of the rollout is characterized by finding the appropriate rollout strategy. The project can begin in phases or all at once. A test scenario can validate the proper rollout. As the final proof of concept, a pilot rollout should be considered.

Consider business hours and critical applications. Minimize the risk by taking controllable steps. The rollout can include down time for the production system. Therefore, timing must be negotiated with all concerned people. The rollout can take place inside or outside of business hours. Remember that some of the required prerequisites may not be available at these times.

A project planning tool is helpful. Lists of resources, availability of resources, time restrictions, dependencies, and cost factors should be documented. Provide a calendar of activities, black-out dates, and milestones.

Chapter 3. High availability example solutions

As companies increase their dependency on technology to deliver services and to process information, the risk of adverse consequences to earnings or capital from operational failures increases. If technology-related problems prevent a company from assessing its data, it may not be able to make payments on schedule, which would severely affect its business. Costly financial penalties may incur, the company's reputation may suffer, and customers may not be able to deliver services or process information of their own. The company may even go out of business all together.

Technology-related problems can increase:

- **Transaction risk**: This arises from problems with service or product delivery
- **Strategic risk**: This results from adverse business decisions or improper implementation of those decisions
- **Reputation risk**: This has its source in negative public opinion
- **Compliance risk**: This arises from violations of laws, rules, regulations, prescribed practices, or ethical standards

Every customer has their own unique characteristics. The implementation of a high availability solution is customized to customer needs and budgets, while keeping in mind the risks of encountering and recovering from problems. This chapter describes examples of customer requirements and the high availability solutions they choose.

3.1 A high availability customer: Scenario 1

A Danish customer with 3,000 users on a large AS/400 system had difficulty finding time to complete tasks that required a dedicated system for maintenance. These tasks included such operations as performing nightly backups, installing new releases, and updating the hardware.

One reason this challenge occurred is because the customer′s AS/400 system was serving all of their retail shops across Scandinavia. As these shops extended their hours, there was less and less time for planned system outages.

They solved their problem by installing mirroring software on two AS/400 systems.

To increase availability, the customer bought a second AS/400 system and connected the two machines with OptiConnect/400. Next, they installed mirroring software, and mirrored everything on the production machine to the backup machine. In the event of a planned or an unplanned system outage, the system users could switch to the backup machine in minutes.

This solution made it easy for the shops to expand their hours and improve sales without losing system availability or sacrificing system maintenance.

3.2 A large financial institution: Scenario 2

Financial services typically require the most highly available solutions. Risks affect every aspect of banking, from interest rates the bank charges to the computers that process bank data. All banks want to avoid risk, but the risk of equipment failure and human error is possible in all systems. This risk may result from sources both within and beyond the bank's control.

Complete details of and a general outline of the SCAAA Bank's new hot backup configuration are provided in the following list:

- An original production AS/400e Model 740 in Madison
- A Model 730 backup AS/400e at IBM's Chicago Business Recovery Services Center
- Both machines running the same level of OS/400 (V4R5)
- High availability software applications transferring data and objects from the customer applications on the production machine to those on the backup machine
- TCP/IP communications protocol used for communications with the BRS Center
- A T1 line from the Madison branch to IBM's POP server in Chicago, where IBM is responsible for communications from the POP server to the BRS center

SAAA Bank of Madison, Wis., is no exception. SAAA specializes in commercial and institutional banking, and it provides comprehensive financial services to other banks, governments, corporations, and organizations. It has received AAA, Aaa and AAA ratings from Standard and Poor's, Moody's, and IBCA (Europe's leading international credit rating agency), respectively. It has offices around the world, with over 5,200 employees woking in commercial centers worldwide.

The Madison branch provides services to U.S. companies and subsidiaries of German corporations in North America. The Information Systems department of the Madison branch is intimately involved with the actual business of the bank, not just its computers.

On a typical day, the workload consists of foreign exchange transactions, money market transactions, securities transactions, options transactions, and loans. The IS department sees every transaction that goes through and does the utmost to ensure that each one executes accurately and promptly. Given the profitability of the Madison branch, the IS department plays a major role in the success of the bank as a whole.

The core data is largely stored on an IBM AS/400e system. Protecting the bank's data means eliminating all single points of failure on that platform. Prior to 1997, the bank took the following steps to ensure business continuity:

- Running regular backups, even making midday backups
- Adding a redundant token-ring card to prevent system failure
- Eliminating cabling problems by using an intelligent hub
- Providing dual air conditioning systems to address environmental variables

- Installing universal power supplies (UPSs) to permit system operations during electrical failure
- Providing RAID5 to maintain parity information across multiple disks
- Installing a second RAID controller to manage the array when the original controller fails
- Leasing twenty seats at the IBM Business Recovery Services (BRS) Center in Sterling Forest, NY, so that the essential work of the bank can be carried out in the event of a disaster

After recognizing the bank's exposure to possible data loss, the bank chose MIMIX Availability Management Software. Running high availability software over a WAN to the BRS Center protects from all risks of data loss.

The high availability software solution at SAAA Bank was modified to meet the demands of the bank's daily schedule. This adjustment was necessary because the applications required journaling to be turned off at the end of the day to facilitate close-of-business processing. This is a characteristic that makes them somewhat difficult to replicate. The bank accomplished this through an elaborate process.

During the day, the bank turned on AS/400e journaling capabilities, but at the end of the business day, the bank shut down MIMIX and the journal files for both systems. At about 6:30 p.m., the bank ran close-of-business processing on both machines simultaneously. This way their databases remained synchronized. About two hours later, when the processing was complete, the bank restarted journaling on both machines and MIMIX was brought back up. The next morning, high availability software started replicating the day's transactions. The entire process was automated, but an operator was always available in case of an emergency. The flexibility of MIMIX enabled the bank to keep the databases in synch at all times.

In 1998, a potentially serious incident occurred at the bank. An operator working on the test system meant to restore some data to the test libraries. Unfortunately, they were copied and loaded into the production system. This overwrote all payments that had to be sent out later that afternoon, leaving the bank in a critical position.

After detecting the error, the operator immediately switched over to the mirrored system on the backup machine located at the BRS center. That system held a mirrored, real time copy of all the bank's transactions for the day. By clicking a button, the operator initiated a full restore from the backup system to the production system, ensuring synchronization of the two databases. This process required only about 25 minutes.

Without high availability application software (MIMIX) the bank would have had to restore from a mid-day backup (which requires about 10 hours worth of work). In addition, about 20 bank staff members would have had to return to the bank and re-enter all the lost transactions at overtime rates. The estimated costs of these activities was around $28K. Even greater costs would have resulted from the one day interest penalties for late or nonexistent outgoing payments.

The fastest and most comprehensive way to ensure maximum uptime and high availability is through added redundancy and fault-tolerant models. With the

implementation of high availability software (MIMIX) at the BRS Center, the bank achieved the highest level of disaster prevention and continuous operations in over 20 years. The strategy provided the availability management, flexibility, and reliability needed to maintain business continuity in the event of a disaster at the location.

3.2.1 Benefits

Looking back, SAAA Bank found that using high availability management software at the BRS center provided these benefits:

- Maintained uninterrupted business processing, despite operator errors and environmental disasters, such as floods, fires, powers outages, and terrorist acts
- Dramatically reduced the cost of disasters, especially disasters that arise at the user level
- Sustained the bank's reputation for reliability: Because the bank acts as a clearing house for several other banks, SAAA Bank had to ensure that the system was operational and that payments were sent out in a timely manner
- Simplified the recovery process: High availability software freed the IS department from lengthy restore projects and eliminated the need to spend time and money re-entering lost data
- Enabled the IS department to switch to the backup database at the BRS center when required

3.3 A large retail company: Scenario 3

Retailing is a new area for very high availability. As retail companies enter the area of e-commerce, the business demands are considerable. Companies operating in a global marketplace must have their systems online 24 hours a day, every day, all year long.

This section describes an example of a large retail company, referred to as EDA Retail for the purposes of the discussion.

EDA Retail is a Danish lumber wholesale business. They also own and run a chain of hardware retail stores across Denmark, Norway, and Sweden, supplying everything for the do-it-yourself person. These hardware stores run a Point-of-Sale (POS) solution from IBM, which connects to a large AS/400 system. At the time this information was gathered, the system was a 12-way model 740 with 1 TB of DASD. All inventory and customer data is stored on this one AS/400 system.

The connection to the AS/400 system is the lifeline to all of the stores. If the connection to the AS/400 system is down, or if the AS/400 system is not up and running at all times, EDA Retail employees cannot take returns, check inventory, check customer data, or perform any related functions while selling goods to their customers.

The unavailability of information developed into a unacceptable problem for EDA Retail. They had grown from a big to a very large enterprise during the four years from when they installed the AS/400 system. Management determined that a breakdown of the AS/400 system would cost the enterprise about 40,000 U.S.

dollars per hour. In addition, planned shutdowns, such as that necessary for the daily backup, system maintenance, upgrades, and application maintenance were becoming a problem. The problem multiplied when a chain of stores in Sweden was taken over and expected to run on the EDA Retail's system. These acquired stores had longer opening hours than the corresponding stores in Denmark, which increased the necessity for long opening hours of the AS/400 system at EDA Retail.

The IBM team did a thorough analysis of EDA Retail's installation in order to develop an environment with no single point of failure. At this time, the IBM team responsible for the customer proposed an extra system for backup, using dedicated communication lines between the two systems.

The proposed solution involved a backup AS/400 system in a new, separate machine room of its own, and an OptiConnect line between the two AS/400 systems. A key feature in this environment was a High Availability solution from Vision Solutions. This solution mirrors all data on the system to a target (backup) system on a real time basis.

In addition to the necessary hardware, this allows EDA Retail to become immune to disasters, planned shutdowns (for example, system maintenance), and unplanned shutdowns (such as system failures). In the case of a shutdown, either planned or unplanned, EDA Retail can switch users to the backup system. Within thirty minutes of the breakdown, operations continue.

The primary test of the project was a *Role Swap*, in which roles are switched between the two systems. The system normally serving as the source system becomes the target system, and vice versa. When the roles are switched and mirroring is started in reverse mode, the user's connection to the new source system is tested to determine if they can run their applications as they normally would. A success is illustrated when the correct data, authorizations, and other objects or information are successfully mirrored, with the transactions also mirrored to the new target system.

The successful completion of a second test of this nature marked the end of the implementation project and the beginning of normal maintenance activities. The users had access to the information they required even while the backup was taken. Their information remained available at all times except during the role swap.

3.4 A small manufacturing company: Scenario 4

This solution applies to manufacturing and to any small business with limited information technology resources. These firms do not typically operate seven days a week, but they may operate 24 hours per day, Monday through Friday.

The business requires that the systems be available when they are needed with a minimum of intervention. If the system fails, they need a backup system on which to run. They can then contact their external I/T services supplier for support. This support can take a day or so to arrive. In the meantime, they run their business with unprotected information.

It is important to note that they are still running in this situation. The third party services provider then arrives, fixes the problem, and re-synchronizes the systems.

DMT Industries is a manufacturer of medical supplies which has become highly globalized over the last several years. Their globalization raises the demand on the IT division to provide a 24 x 7 operation.

DMT Industries has various platforms installed. One of the strategic business applications, which must be available at all times, is running on the AS/400 platform.

A total of 1,500 users from all over the world have access to the system, which is located in Denmark. Until early 1998, DMT Industries could offer only 20 hours of system availability per day. The remaining four hours were used for the nightly backup. As it became necessary to offer 24 hour availability, they decided to implement a high availability solution.

The installation then consisted of two identical AS/400 systems, located in separate machine rooms, a dedicated 100 Mbit Ethernet connection between them, and software from Vision Solutions. This enabled DMT Medical to perform their nightly backup on the backup system to which all production data was mirrored. They simply stopped the mirrored data from the production machine from being applied on the backup system while the backup ran. The users could continue operations on the production system as required.

3.5 A distribution company: Scenario 5

This example represents a three-tiered SAP solution (reportedly the largest site worldwide).

TVBCo is a distribution company utilizing J.D. Edwards (JDE) applications. The company focuses on expanding knowledge about human protein molecules and transforming them by means of biotechnological methods and clinical tests. TVBCo had developed into a completely integrated worldwide pharmaceutical business. Its European Distribution Center in Holland houses a central IBM AS/400e production system, accessed by a large number of TVBCos branches in Europe. A second AS/400e machine is used for testing, development, and backup. The production AS/400e system plays a critical role in TVBCos European operations, so the company wanted to reduce its downtime to a minimum.

In cooperation with IBM, TVBCo started developing an availability management plan and purchased two AS/400e systems. MIMIX availability management software solution was installed.

High availability software immediately replicated all production system transactions on a defined backup system and performed synchronization checking to verify data integrity. Critical data and other items were always present and available on the backup system so that users could switch to that system and continue working if an unplanned outage occured.

Aside from the immediate safekeeping of all data, high availability software maintained an accurate, up-to-date copy of the system setup on the back-up machine. For that purpose, all user profiles, subsystem descriptions, and other

objects were replicated. High availability software also controlled the production system and the actual switchover in case of a failure. The backup machine monitored whether the production machine was still on. The moment it detected that something was wrong, it warned the system operator, who could decide to switch over. When that decision was made, all necessary actions had already been set out in a script so that the operator only had to intervene in exceptional cases.

The TVBCo customized procedure included sending warnings to all users, releasing the backup database, activating subsystems and backup users, and switching interactive users to the backup system.

TVBCo soon decided to start using high availability software as its worldwide standard for processing and invoicing. The data changed continuously, and a traditional backup could not be produced because too few stable points existed in the day. With the chosen high availability solution, the backup was made while the file was active. Therefore, taking the application offline was not necessary. In addition, the high availability solution backup ran, on average, only a couple of seconds (sometimes even less) behind production. Contrast this if a tape were made every hour (the backup could have run up to one hour behind in real time).

The second application, an electronic data interchange (EDI) package, automates the purchase of supplies. This package retrieves batch data from the computer networks managed by the company's suppliers. This process is not continuous so a more traditional backup method may be suitable. However, after estimating the considerable effort required to adapt the batch protocols and DDM files, it was decided to use high availability software. Because their chosen high availability software solution kept the backup data current (almost to the second), the company did not need to request that the batches be sent again when a problem occured with the production machine. Also, high availability software allowed the backup to be made in batches (for example, to relieve the network during peak hours).

The third critical application replicated by high availability software is a DSI logistics program. Since this application not only generates packing lists and is used for updating the contents of the warehouse in the J.D. Edwards database, it must have information about the most recent setup of the warehouse. This information must, of course, remain available when the production equipment breaks down. Otherwise, the orders may be processed but the forklift operators would not know where to unload the merchandise. Like the J.D. Edwards application, the input and output of this application (and, therefore, the replicating) is a real time process.

The data from the J.D. Edwards application and the IBM application are copied to the backup while the file is active. The batch processes from a variety of platforms, are copied to the AS/400e production system at scheduled intervals. With high availability software, it is not necessary to take the application offline because the backup is made while the file is active. High availability software also controls the production system and actual switchover in case of a failure.

TVBCos data center is supported by a staff of 70. The hardware consists of two AS/400e Model 500 systems connected with TCP on ethernet adapters.

Part 2. AS/400 high availability functions

Many areas of a system implementation contribute to the availability rating. Each option provides differing degrees of availability.

Part II discusses the hardware, storage options, operating system features, and network components that contribute to a high availability solution. Appendix F, “Cost components of a business case” on page 169, provides a reference to compare the availability options of journaling, mirrored protection, device parity protection, and others.

Chapter 4. Hardware support for single system high availability

The selection of hardware components provides a basis for system availability options. This chapter discusses some of the considerations for protecting your data and available features and tools from a hardware perspective on a single system and options using multiple systems.

Hardware selections greatly contribute to a system's high availability characteristics. This chapter discusses:

- Data protection options
- Concurrent maintenance
- Hot spares
- OptiConnect
- Clusters
- LPAR
- Power options
- Tape devices

4.1 Protecting your data

Your first defense against data loss is a good backup and recovery strategy. You need a plan for regularly saving the information on your system. In addition to having a working backup and recovery strategy, you should also employ some form of data protection on your system.

When you think about protecting your system from data loss, make the following considerations:

- **Recovery**: Can you retrieve the information that you lost, either by restoring it from backup media or by creating it again?
- **Availability**: Can you reduce or eliminate the amount of time that your system is unavailable after a problem occurs?
- **Serviceability**: Can you service it without affecting the data user?

Disk protection can help prevent data loss and keep your system from stopping if you experience a disk failure.

Remember

Although disk protection can reduce downtime or make recovery faster, it is not a replacement for regular backups. Disk protection cannot help you recover from a complete system loss, a processor failure, or a program failure.

The topics that follow provide information on the different types of disk protection, as well as using the types with one another.

4.2 Disk protection tools

Several disk availability tools are available for reducing or eliminating system downtime. They also help with data recovery after a disk failure. The tools include:

- Mirrored protection
- Device parity protection
- Auxiliary storage pools (ASPs)
- Others

These disk protection methods help protect your data. You can use these methods in different combinations with one another. Your choice of disk tools determines your level of disk protection and vice versa. Figure 6 shows the different levels of availability.

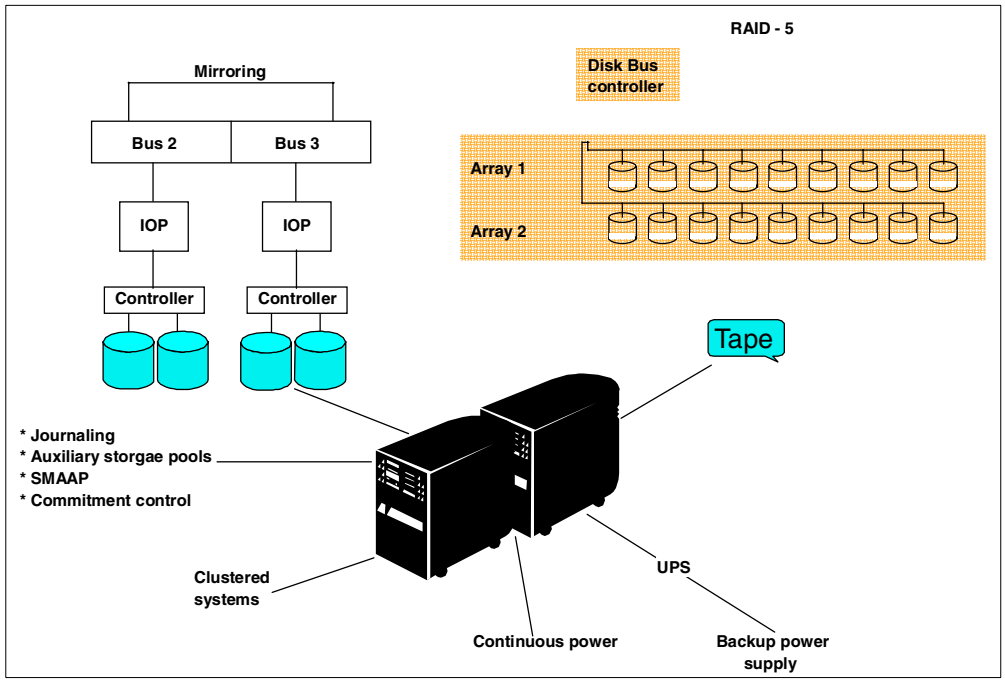


Figure 6. Levels of availability

High availability, as shown in Figure 6, includes mirroring, RAID-5, offline backup, UPS, CPM, clustering, journaling, auxiliary storage pools, SMAPP, commitment control, and save-while-active components.

Topics not covered in this Redpaper are discussed in *The System Administrator's Companion to AS400 Availability and Recovery*, SG24-2161, and the *Backup and Recovery*, SC41-5304.

4.3 Disk mirroring

Mirrored protection is an OS/400 high availability function in the licensed internal code that protects data from loss due to failure, or due to damage to a disk-related component.

Mirrored protection is a high availability software function that duplicates disk-related hardware components to keep the AS/400 system available if one of the disk components fails. It prevents a loss of data in case of a disk-related hardware failure. Mirroring is used on any model of the AS/400 system and is a part of the Licensed Internal Code (LIC).

Different levels of mirrored protection are possible, depending on what hardware is duplicated. The hardware components that can be duplicated include:

- Disk units (to provide the lowest (relative) level of availability)
- Disk controllers
- Disk I/O processors (IOP)
- Buses (to provide the highest (relative) level of availability)

Mirroring protection is configured by the ASP. For optimum protection, there must be an even number of components at each level of protection. The system remains available during a disk, controller, IOP, or bus failure if the failing component and hardware components that are attached to it are duplicated.

When you start mirrored protection or add disk units to an ASP that has mirrored protection, the system creates mirrored pairs using disk units that have identical capacities. Data is protected because the system keeps two copies of data on two separate disk units.

When a disk-related component fails, the system can continue to operate without interruption by using the mirrored copy of the data until the failed component is repaired. The overall goal is to protect as many disk-related components as possible. To provide maximum hardware redundancy and protection, the system attempts to pair disk units that are attached to different controllers, input/output processors, and buses.

If you have a multi-bus system or a large single-bus system, consider using mirrored protection. The greater the number of disk units attached to a system, the more frequently disk-related hardware failures occur. This is because there are more individual pieces of hardware that can fail. Therefore, the possibility of data loss or loss of availability as a result of a disk or other hardware failure is more likely.

Also, as the amount of disk storage on a system increases, the recovery time after a disk storage subsystem hardware failure increases significantly. Downtime becomes more frequent, more lengthy, and more costly.

Mirroring can be used on any AS/400 system model. The system remains available during a failure if a failing component and the hardware components that are attached to it have been duplicated. See 4.8, “Concurrent maintenance” on page 54, to better understand the maintenance aspect.

Remote mirroring support allows you to have one mirrored unit within a mirrored pair at the local site, and the second mirrored unit at a remote site.

For some systems, standard DASD mirroring remains the best option. Evaluate the uses and needs of your system, consider the advantages and disadvantages of each type of mirroring support, and decide which is best for you.

Refer to B.2.3, “Mirrored protection: How it works” on page 151, for a more detailed description.

4.3.1 Standard mirrored protection

Standard DASD mirroring support requires that both disk units of the load source mirrored pair (unit 1) are attached to the multi-function I/O processor (MFIOP). This option allows the system to initial program load (IPL) from either load source

in the mirrored pair. The system can dump main storage to either load source if the system terminates abnormally. However, since both load source units must be attached to the same I/O processor (IOP), controller level protection is the best mirroring protection possible for the load source mirrored pair.

How the AS/400 system addresses storage

Disk units are assigned to an auxiliary storage pool (ASP) on a unit basis. The system treats each storage unit within a disk unit as a separate unit of auxiliary storage. When a new disk unit is attached to the system, the system initially treats each storage unit within as non-configured. Through Dedicated Service Tools (DST) options, you can add these nonconfigured storage units to either the system ASP or a user ASP of your choice. When adding non-configured storage units, use the serial number information that is assigned by the manufacturer to ensure that you are selecting the correct physical storage unit. Additionally, the individual storage units within the disk unit can be identified through the address information that can be obtained from the DST Display Disk Configuration display.

When you add a nonconfigured storage unit to an ASP, the system assigns a unit number to the storage unit. The unit number can be used instead of the serial number and address. The same unit number is used for a specific storage unit even if you connect the disk unit to the system in a different way.

When a unit has mirrored protection, the two storage units of the mirrored pair are assigned the same unit number. The serial number and the address distinguish between the two storage units in a mirrored pair. To determine which physical disk unit is being identified with each unit number, note the unit number assignment to ensure correct identification. If a printer is available, print the DST or SST display of your disk configuration. If you need to verify the unit number assignment, use the DST or SST Display Configuration Status display to show the serial numbers and addresses of each unit.

The storage unit that is addressed by the system as Unit 1 is always used by the system to store licensed internal code and data areas. The amount of storage that is used on Unit 1 is quite large and varies depending on your system configuration. Unit 1 contains a limited amount of user data. Because Unit 1 contains the initial programs and data that is used during an IPL of the system, it is also known as the load source unit.

The system reserves a fixed amount of storage on units other than Unit 1. The size of this reserved area is 1.08 MB per unit. This reduces the space available on each unit by that amount.

4.3.2 Mirrored protection: Benefits

With the best possible mirrored protection configuration, the system continues to run after a single disk-related hardware failure. On some system units, the failed hardware can sometimes be repaired or replaced without having to power down the system. If the failing component is one that cannot be repaired while the system is running, such as a bus or an I/O processor, the system usually continues to run after the failure. Maintenance can be deferred and the system can be shut down normally. This helps to avoid a longer recovery time.

Even if your system is not a large one, mirrored protection can provide valuable protection. A disk or disk-related hardware failure on an unprotected system leaves your system unusable for several hours. The actual time depends on the

kind of failure, the amount of disk storage, your backup strategy, the speed of your tape unit, and the type and amount of processing the system performs.

4.3.3 Mirrored protection: Costs and limitations

The main cost of using mirrored protection lies in additional hardware. To achieve high availability, and prevent data loss when a disk unit fails, you need mirrored protection for all the auxiliary storage pools (ASPs). This usually requires twice as many disk units. If you want continuous operation and data loss prevention when a disk unit, controller, or I/O processor fails, you need duplicate disk controllers as well as I/O processors.

A model upgrade can be done to achieve nearly continuous operation and to prevent data loss when any of these failures occur. This includes a bus failure. If Bus 1 fails, the system cannot continue operation. Because bus failures are rare, and bus-level protection is not significantly greater than I/O processor-level protection, you may not find a model upgrade to be cost-effective for your protection needs.

Mirrored protection has a minimal reduction in system performance. If the buses, I/O processors, and controllers are no more heavily loaded on a system with mirrored protection than they would be on an equivalent system without mirrored protection, the performance of the two systems should be approximately the same.

In deciding whether to use mirrored protection on your system, evaluate and compare the cost of potential downtime against the cost of additional hardware over the life of the system. The additional cost in performance or system complexity is usually negligible.

Also consider other availability and recovery alternatives, such as device parity protection. Mirrored protection normally requires twice as many storage units. For concurrent maintenance and higher availability on systems with mirrored protection, other disk-related hardware may be required.

Limitations

Although mirrored protection can keep the system available after disk-related hardware failures occur, it is not a replacement for save procedures. There can be multiple types of disk-related hardware failures, or disasters (such as flood or sabotage) where recovery requires backup media.

Mirrored protection cannot keep your system available if the remaining storage unit in the mirrored pair fails before the first failing storage unit is repaired and mirrored protection is resumed.

If two failed storage units are in different mirrored pairs, the system is still available. Normal mirrored protection recovery is done because the mirrored pairs are not dependent on each other for recovery. If a second storage unit of the same mirrored pair fails, the failure may not result in a data loss. If the failure is limited to the disk electronics, or if the service representative can successfully use the Save Disk Unit Data function to recover all of the data (a function referred to as “pump”), no data is lost.

If both storage units in a mirrored pair fail causing data loss, the entire ASP is lost and all units in the ASP are cleared. Be prepared to restore your ASP from the backup media and apply any journal changes to bring the data up to date.

4.3.4 Determining the level of mirrored protection

The level of mirrored protection determines whether the system continues running when different levels of hardware fails. The level of protection is the amount of duplicate disk-related hardware that you have. The more mirrored pairs that have higher levels of protection, the more often your system is usable when disk-related hardware fails. You may decide that a lower level of protection is more cost-effective for your system than a higher level.

The four levels of protection, from lowest to highest, are as follows:

- Disk unit-level protection
- Controller-level protection
- Input/output processor-level protection
- Bus-level protection

When determining what level of protection is adequate, consider the relative advantages of each level of protection with respect to the following considerations:

- The ability to keep the system operational during a disk-related hardware failure.
- The ability to perform maintenance concurrently with system operations. To minimize the time that a mirrored pair is unprotected after a failure, you may want to repair failed hardware while the system is operating.

During the *start mirrored protection* operation, the system pairs the disk units to provide the maximum level of protection for the system. When disk units are added to a mirrored ASP, the system pairs only those disk units that are added without rearranging the existing pairs. The hardware configuration includes the hardware and how the hardware is connected.

The level of mirrored protection determines whether the system continues running when different levels of hardware fail. Mirrored protection always provides disk unit-level protection that keeps the system available for a single disk unit failure. To keep the system available for failures of other disk-related hardware requires higher levels of protection. For example, to keep the system available when an I/O processor (IOP) fails, all of the disk units attached to the failing IOP must have mirrored units attached to different IOPs.

The level of mirrored protection also determines whether concurrent maintenance can be done for different types of failures. Certain types of failures require concurrent maintenance to diagnose hardware levels above the failing hardware component. An example would be diagnosing a power failure in a disk unit that requires resetting the I/O processor to which the failed disk unit is attached. In this case, IOP-level protection is required.

The level of protection you get depends on the hardware you duplicate. If you duplicate disk units, you require disk unit-level protection. If you also duplicate disk unit controllers, you require controller-level protection. If you duplicate

input/output processors, you require IOP-level protection. If you duplicate buses, you require bus-level protection.

Mirrored units always have, at least, disk unit-level protection. Because most internal disk units have the controller packaged along with the disk unit, they have at least controller-level protection.

4.3.4.1 Disk unit-level protection

All mirrored storage units have a minimum of disk unit-level protection if they meet the requirements for starting mirrored protection (storage units are duplicated). If your main concern is protecting data and *not* high availability, disk unit-level protection may be adequate.

The disk unit is the most likely hardware component to fail, and disk unit-level protection keeps your system available after a disk unit failure. Concurrent maintenance is often possible for certain types of disk unit failures with disk unit-level protection.

Figure 7 shows disk unit-level protection. The two storage units make a mirrored pair. With disk unit-level protection, the system continues to operate during a disk unit failure. If the controller or I/O processor fails, the system cannot access data on either of the storage units of the mirrored pair, and the system is unusable.

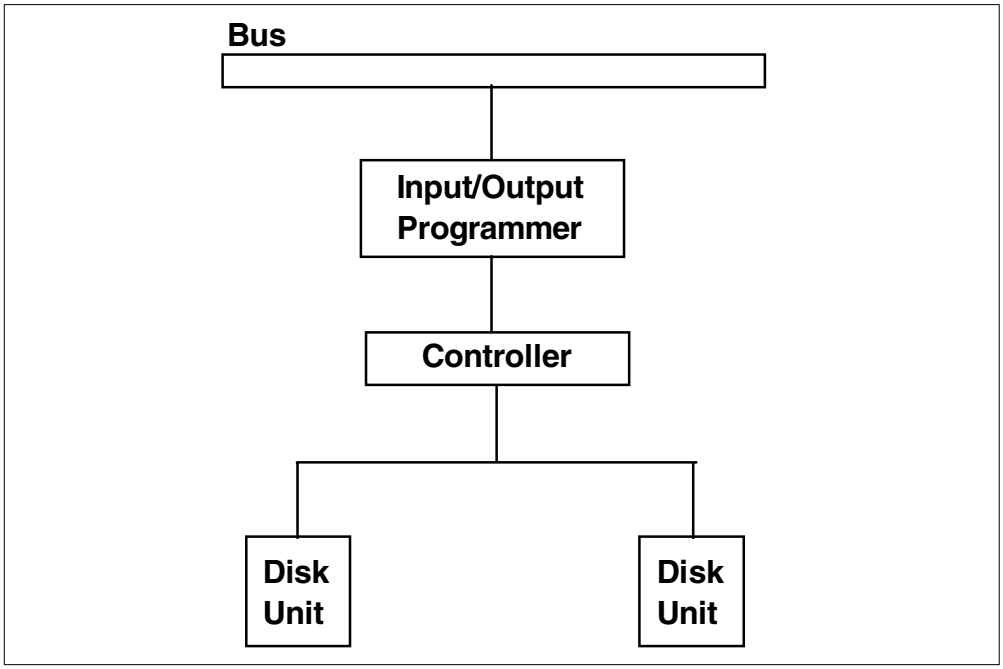


Figure 7. Disk unit level protection

4.3.4.2 Controller-level protection

If the planned disk units do not require a separate controller, you already have controller-level protection for as many units as possible and do not need to do anything else. If your planned disk units *do* require a separate controller, add as many controllers as possible while keeping within the defined system limits. Then balance the disk units among them according to the standard system configuration rules.

To keep your system available when a controller fails, consider using concurrent maintenance. The controller must be dedicated to the repair action in this process. If any disk units attached to the controller do not have controller-level protection, concurrent maintenance is not possible.

To achieve controller-level protection, all disk units must have a mirrored unit attached to a different controller. Most internal disk units have their controller packaged as part of the disk unit, so internal disk units generally have at least controller-level protection. Use problem recovery procedures in preparation for isolating a failing item or to verify a repair action.

Figure 8 illustrates controller-level protection.

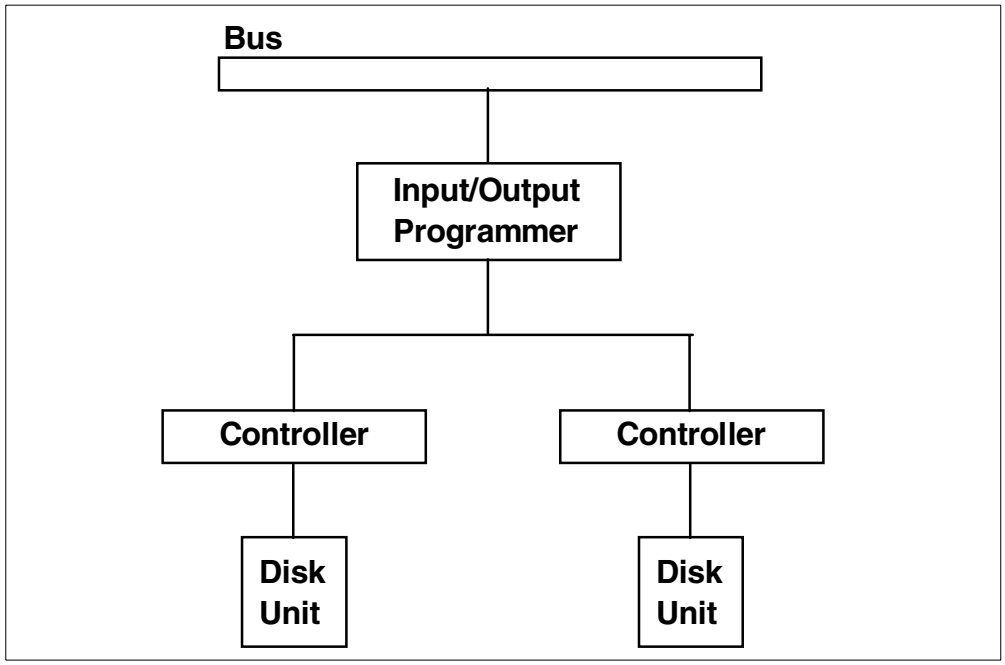


Figure 8. Input/output controller-level protection

The two storage units make a mirrored pair. With controller-level protection, the system can continue to operate after a disk controller failure. If the I/O processor fails, the system cannot access data on either of the disk units, and the system is unusable.

4.3.4.3 IOP-level protection

If you want IOP-level protection and you do not already have the maximum number of IOPs on your system, add as many IOPs as possible while keeping within the defined system limits. Then, balance the disk units among them according to the standard system configuration rules. You may need to add additional buses to attach more IOPs.

To achieve I/O processor-level protection, all disk units that are attached to an I/O processor must have a mirrored unit attached to a different I/O processor. On many systems, I/O processor-level protection is not possible for the mirrored pair for Unit 1.

IOP-level protection is useful to:

- Keep your system available when an I/O processor fails
- Keep your system available when the cable attached to the I/O processor fails
- Concurrently repair certain types of disk unit failures or cable failures. For these failures, concurrent maintenance needs to reset the IOP. If any disk units that are attached to the IOP do not have IOP-level protection, concurrent maintenance is not possible.

Figure 9 illustrates IOP-level protection.

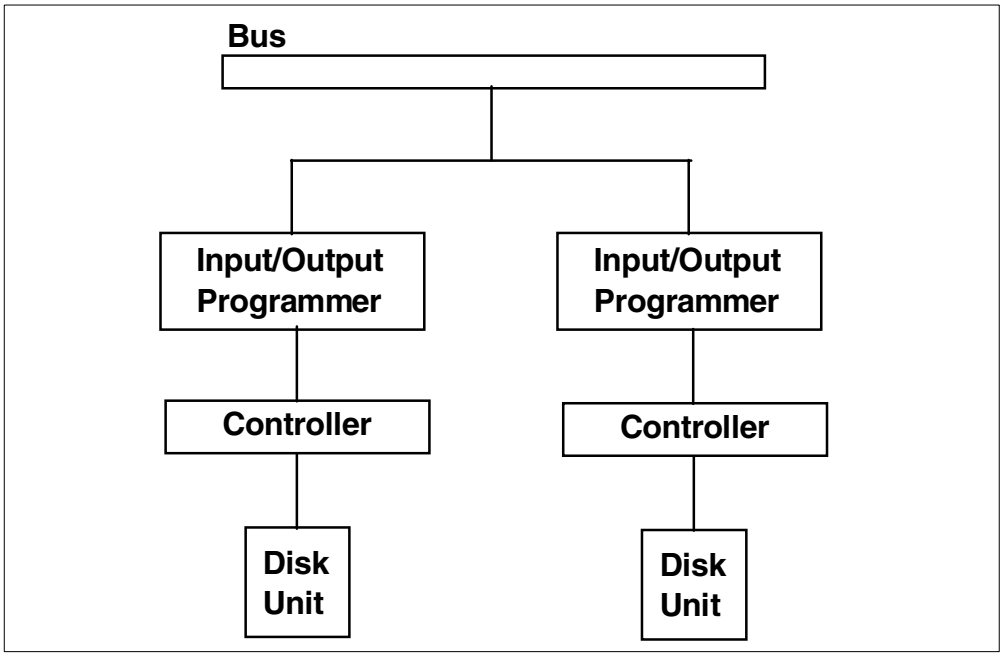


Figure 9. IOP-level protection

The two storage units make a mirrored pair. With IOP-level protection, the system continues to operate after an I/O processor failure. The system becomes unusable only if the bus fails.

4.3.4.4 Bus-level protection

If you want bus-level protection, and you already have a multiple-bus system, nothing must be done. If your system is configured according to standard configuration rules, the mirrored pairing function pairs up storage units to provide bus-level protection for as many mirrored pairs as possible. If you have a single-bus system, you can add additional buses as a feature option on systems supporting multiple buses.

Bus-level protection can allow the system to run when a bus fails. However, bus-level protection is often not cost-effective because of the following problems:

- If Bus 1 fails, the system is not usable.
- If a bus fails, disk I/O operations may continue, but so much other hardware is lost (such as work stations, printers, and communication lines) that, from a practical standpoint, the system is not usable.

- Bus failures are rare compared with other disk-related hardware failures.
- Concurrent maintenance is not possible for bus failures.

To achieve bus-level protection, all disk units that are attached to a bus must have a mirrored unit attached to a different bus. Bus-level protection is not possible for Unit 1.

Figure 10 illustrates bus-level protection.

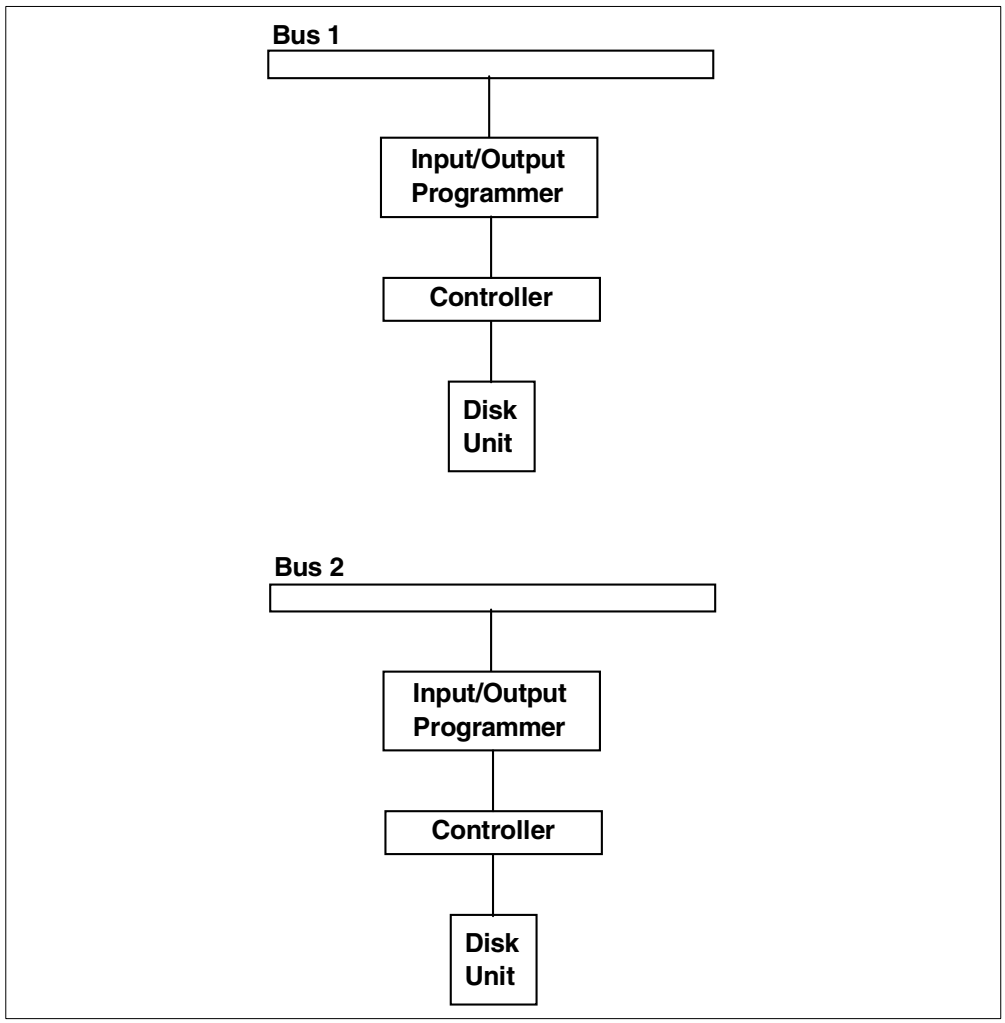


Figure 10. Bus-level protection

The two storage units make a mirrored pair. With bus-level protection, the system continues to operate after a bus failure. However, the system cannot continue to operate if Bus 1 fails.

4.3.5 Determining the hardware required for mirroring

To communicate with the rest of the system, disk units are attached to controllers, which are attached to I/O processors, which are attached to buses. The number of each of these types of disk-related hardware that are available on the system directly affects the level of possible protection.

To provide the best protection and performance, each level of hardware should be balanced under the next level of hardware. That is, the disk units of each device type and model should be evenly distributed under the associated controllers. The same number of controllers should be under each I/O processor for that disk type. Balance the I/O processors among the available buses.

To plan what disk-related hardware is needed for your mirrored system, plan the total number and type of disk units (old and new) that are needed on the system, as well as the level of protection for the system. It is not always possible to plan for and configure a system so that all mirrored pairs meet the planned level of protection. However, it is possible to plan a configuration in which a very large percentage of the disk units on the system achieve the desired level of protection.

When planning for additional disk-related hardware, perform the following steps:

1. Determine the minimum hardware that is needed for the planned disk units to function. Plan for one disk unit size (capacity) at a time.
2. Plan the additional hardware needed to provide the desired level of protection for each disk unit type

Planning the minimum hardware needed to function
Various rules and limits exist on how storage hardware can be attached. The limits may be determined by hardware design, architecture restrictions, performance considerations, or support concerns.

For each disk unit type, first plan for the controllers that are needed and then for the I/O processors that are needed. After planning the number of I/O processors needed for all disk unit types, use the total number of I/O processors to plan for the number of buses that are needed.

4.3.6 Mirroring and performance

When mirrored protection is started, most systems show little difference in performance. In some cases, mirrored protection can improve performance. Generally, functions that mostly perform read operations experience equal or better performance with mirrored protection. This is because read operations have a choice of two storage units to read from. The unit with the faster expected response time is selected. Operations that mostly perform write operations (such as updating database records) may see slightly reduced performance on a system that has mirrored protection because all changes must be written to both storage units of the mirrored pair. Therefore, restore operations are slower.

In some cases, if the system ends abnormally, the system cannot determine whether the last updates were written to both storage units of each mirrored pair. If the system is not sure that the last changes were written to both storage units of the mirrored pair, the system synchronizes the mirrored pair by copying the data in question from one storage unit of each mirrored pair to the other storage unit.

The synchronization occurs during the IPL that follows the abnormal system end. If the system can save a copy of main storage before it ends, the synchronization process takes just a few minutes. If not, the synchronization process can take much longer. The extreme case could be close to a complete synchronization.

If you have frequent power outages, consider adding an uninterruptible power supply (UPS) to your system. If main power is lost, the UPS allows the system to continue. A basic UPS supply provides the system with enough time to save a copy of main storage before ending. This prevents a long recovery. Both storage units of the load source mirrored pair must be powered by the basic UPS.

4.3.7 Determining the extra hardware required for performance

Mirrored protection normally requires additional disk units and input/output processors. However, in some cases, you may need additional hardware to achieve the level of desired performance.

Use these points to decide how much extra hardware you may need:

- Mirrored protection causes a minor increase in central processing unit usage (approximately 1% to 2%).
- Mirrored protection requires storage in the machine pool for general purposes and for each mirrored pair. If you have mirrored protection, increase the size of your machine pool by approximately 12 KB for each 1 GB of mirrored disk storage (12 KB for 1 GB DASD, 24 KB for 2 GB DASD, etc.).
- During synchronization, mirrored protection uses an additional 512 KB of memory for each mirrored pair being synchronized. The system uses the pool with the most storage.
- To maintain equivalent performance after starting mirrored protection, your system should have the same ratio of disk units to I/O processors as it did before. To add I/O processors, you may need to upgrade your system for additional buses.

  Note: Because of the limit on buses and I/O processors, you may not be able to maintain the same ratio of disk units to I/O processors. In this case, system performance may degrade.

4.4 Remote DASD mirroring support

Standard DASD mirroring support requires that both disk units of the load source mirrored pair (Unit 1) are attached to the Multi-function I/O Processor (MFIOP). This allows the system to IPL from either load source in the mirrored pair and allows the system to dump main storage to either load source if the system ends abnormally. However, since both load sources must be attached to the same I/O Processor (IOP), the best mirroring protection possible for the load source mirrored pair is controller-level protection. To provide a higher level of protection for your system, use remote load source mirroring and remote DASD mirroring.

Remote DASD mirroring support, when combined with remote load source mirroring, mirrors the DASD on local optical buses with the DASD on optical buses that terminate at a remote location. In this configuration, the entire system, including the load source, can be protected from a site disaster. If the remote site is lost, the system can continue to run on the DASD at the local site. If the local DASD and system unit are lost, a new system unit can be attached to the set of DASD at the remote site, and system processing can be resumed.

Remote DASD mirroring, like standard DASD mirroring, supports mixing device-parity-protected disk units in the same ASP with mirrored disk units. The device parity DASD can be located at either the local or the remote site. However,

if a site disaster occurs at the site containing the device parity DASD, all data in the ASPs containing the device parity DASD is lost.

Remote mirroring support makes it possible to divide the disk units on your system into a group of local DASD and a group of remote DASD. The remote DASD is attached to one set of optical buses and the local DASD to another set of buses. The local and remote DASD can be physically separated from one another at different sites by extending the appropriate optical buses to the remote site.

4.4.1 Remote load source mirroring

Remote load source mirroring support allows the two disk units of the load source to be on different IOPs or system buses. This provides IOP- or bus-level mirrored protection for the load source. However, in such a configuration, the system can only IPL from, or perform a main storage dump to, the load source attached to the MFIOP. If the load source on the MFIOP fails, the system can continue to run on the other disk unit of the load source mirrored pair, but the system is not able to IPL or perform a main storage dump until the load source attached to the MFIOP is repaired and usable.

4.4.2 Enabling remote load source mirroring

To use remote load source mirroring support, remote load source mirroring must first be enabled. Mirrored protection must then be started for ASP 1. If remote load source mirroring support is enabled after mirrored protection has already been started for ASP 1, the existing mirrored protection and mirrored pairing of the load source must not change. Remote load source mirroring support can be enabled in either the DST or the SST environment. If you attempt to enable remote load source mirroring, and it is currently enabled, the system displays a message that remote load source mirroring is already enabled. There are no other errors or warnings for enabling remote load source mirroring support.

If the remote load source is moved to the MFIOP, the IOP and system may not recognize it because of the different DASD format sizes used by different IOPs. If the remote load source is missing after it has been moved to the MFIOP, use the DST Replace disk unit function to replace the missing load source with itself. This causes the DASD to be reformatted so that the MFIOP can use it. The disk unit is then synchronized with the active load source.

Remote load source mirroring may be disabled from either DST or SST. However, disabling remote load source mirroring is not allowed if there is a load source disk unit on the system that is not attached to the MFIOP. If you attempt to disable remote load source mirroring support and it is currently disabled, the system displays a message that remote load source mirroring is already disabled.

4.4.3 Using remote load source mirroring with local DASD

Remote load source mirroring can be used to achieve IOP-level or bus-level protection of the load source mirrored pair, even without remote DASD or buses on the system. There is no special setup required, except for ensuring that a disk unit of the same capacity as the load source is attached to another IOP or bus on the system. To achieve bus-level protection of all mirrored pairs in an ASP, configure your system so that no more than one-half of the DASD of any given capacity in that ASP are attached to any single bus. To achieve IOP-level

protection of all mirrored pairs in an ASP, have no more than one-half of the DASD of any given capacity in the ASP attached to any single IOP.

There is no special start mirroring function for remote load source support. The system detects that remote load source mirroring is enabled and automatically pairs up disk units to provide the best level of possible protection. It is not possible to override or influence the pairing of the disk units other than by changing the way the hardware of the system is connected and configured. Normal mirroring restrictions that concern total ASP capacity, an even number of disk units of each capacity, and other such considerations, apply.

4.4.3.1 Remote DASD mirroring: Advantages

Advantages of remote DASD mirroring include:

- Providing IOP-level or bus-level mirrored protection for the load source
- Allowing the DASD to be divided between two sites, mirroring one site to another, to protect against a site disaster

4.4.3.2 Remote DASD mirroring: Disadvantages

Disadvantages of remote DASD mirroring include:

- A system that uses Remote DASD Mirroring is only able to IPL from one DASD of the load source mirrored pair. If that DASD fails and cannot be repaired concurrently, the system cannot be IPLed until the failed load source is fixed and the remote load source recovery procedure is performed.
- When Remote DASD Mirroring is active on a system, and the one load source the system can use to IPL fails, the system cannot perform a main storage dump if the system ends abnormally. This means that the system cannot use the main storage dump or continuously-powered main store (CPM) to reduce recovery time after a system crash. It also means that the main storage dump is not available to diagnose the problem that causes the system to end abnormally.

4.5 Planning your mirroring installation

If you decide that remote DASD mirroring is right for your system, prepare your system and then start site-to-site mirroring. Determine whether your system is balanced and meets standard configuration rules. The system must be configured according to the standard rules for the mirrored pairing function to pair up storage units to provide the best possible protection from the hardware that is available. Plan for the new units to add for each ASP.

If you plan to start mirrored protection on a new system, that system is already configured according to standard configuration rules. If you are using an older system, it may not follow the standard rules. However, wait until after attempting to start mirrored protection before reconfiguring any hardware.

When considering mirrored protection, review these planning steps:

1. Decide which ASP or ASPs to protect.
2. Determine the disk storage capacity requirements.
3. Determine the level of protection that is needed for each mirrored ASP.
4. Determine what extra hardware is necessary for mirrored protection.
5. Determine what extra hardware is needed for performance.

In general, the units in an ASP should be balanced across several I/O processors, rather than all being attached to the same I/O processor. This provides better protection and performance. Plan the user ASPs that have mirrored protection and determine what units to add to the ASPs. Refer to Chapter 5, “Auxiliary storage pools (ASPs)” on page 63, for more information about ASPs.

4.5.1 Comparing DASD management with standard and remote mirroring

For the most part, managing DASD with remote mirroring is similar to managing DASD with standard mirroring. The differences are in how you add disk units and how you restore mirrored protection after a recovery.

Adding disk units

Unprotected disk units must be added in pairs just as with general mirroring. To achieve remote protection of all added units, one half of the new units of each capacity of DASD should be in the remote group and one half in the local group. Single device-parity protected units may be added to ASPs using remote mirroring. However, the ASP is not protected against a site disaster.

4.6 Device parity protection

Device parity protection is a high availability hardware function (also known as RAID-5) that protects data from loss due to a disk unit failure or because of damage to a disk. It allows the system to continue to operate when a disk unit fails or disk damage occurs.

The system continues to run in an exposed mode until the damaged unit is repaired and the data is synchronized to the replaced unit. To protect data, the disk controller or input/output processor (IOP) calculates and saves a parity value for each bit of data. Parity protection is built into many IOPs. It is activated for disk units that are attached to those IOPs.

Recommendation

If a failure occurs, correct the problem quickly. In the unlikely event that another disk fails in the same parity set, you may lose data.

Device parity involves calculating and saving a parity value for each bit of data.

Conceptually, the parity value is computed from the data at the same location on each of the other disk units in the device parity set. When a disk failure occurs, the data on the failing unit is reconstructed using the saved parity value and the values of bits in the same location on other disks. The system continues to run while the data is being reconstructed.

Logically, the implementation of device parity protection is similar to the system checksum function. However, device parity is built into the hardware. Checksum, on the other hand, is started or stopped using configuration options on the AS/400 system menu.

Note

System checksum is another disk protection method similar to device parity. Checksum is not supported on RISC systems, and it is not discussed in this Redpaper. You can find information on checksum in *Backup and Recovery*, SC41-5306.

The overall goal of device parity protection is to provide high availability and to protect data as inexpensively as possible.

If possible, protect all the disk units on your system with device parity protection or mirrored protection. This prevents the loss of information when a disk failure occurs. In many cases, you can also keep your system operational while a disk unit is being repaired or replaced.

Remember

Device parity protection is not a substitute for a backup and recovery strategy. Device parity protection can prevent your system from stopping when certain types of failures occur. It can speed up your recovery process for certain types of failures. But device parity protection does not protect you from many types of failures, such as system outages that are caused by failures in other disk-related hardware (for example, disk controllers, disk I/O processors, or a system bus).

Before using device parity protection, note the benefits that are associated with it, as well as the costs and limitations.

Some device parity protection advantages:

- It can prevent your system from stopping when certain types of failures occur.
- It can speed up your recovery process for certain types of failures, such as a site disaster or an operator or programmer error.
- Lost data is automatically reconstructed by the disk controller after a disk failure.
- The system continues to run after a single disk failure.
- A failed disk unit can be replaced without stopping the system.
- Device parity protection reduces the number of objects that are damaged when a disk fails.

Some device parity protection disadvantages:

- It is *not* a substitute for a backup and recovery strategy.
- It does *not* provide protection from all types of failures, such as a site disaster or an operator or programmer error.
- Device parity protection can require additional disk units to prevent slower performance.
- Restore operations can take longer when you use device parity protection.

For information on planning for device parity protection, refer to Appendix B, “Planning for device parity protection” on page 147.

4.6.1 How device parity protection affects performance

Device parity protection requires extra I/O operations to save the parity data. This may cause a performance problem. To avoid this problem, some IOPs contain a non-volatile write cache that ensures data integrity and provides faster write capabilities. The system is notified that a write operation is complete as soon as a copy of the data is stored in the write cache. Data is collected in the cache before it is written to a disk unit. This collection technique reduces the number of physical write operations to the disk unit. Because of the cache, performance is generally about the same on protected and unprotected disk units.

Applications that have many write requests in a short period of time, such as batch programs, can adversely affect performance. A single disk unit failure can adversely affect the performance for both read and write operations.

The additional processing that is associated with a disk unit failure in a device parity set can be significant. The decrease in performance is in effect until the failed unit is repaired (or replaced) and the rebuild process is complete. If device parity protection decreases performance too much, consider using mirrored protection. These topics provide additional details on how a disk unit failure affects performance:

- Disk unit failure in a device parity protection configuration
- Input/output operations during a rebuild process
- Read operations on a failed disk unit
- Write operations on a failed disk unit

4.6.1.1 Disk unit failure in a device parity protection configuration

The write-assist device is suspended when a disk unit failure occurs in a subsystem with device parity protection. If the write-assist device fails, it is not used again until the repair operation is completed. The performance advantage of the write-assist device is lost until the disk unit is repaired.

The subsystems with device parity protection are considered to be exposed until the synchronization process completes after replacing the failed disk unit. While the disk unit is considered exposed, additional I/O operations are required.

4.6.1.2 Input/output operations during a rebuild process

I/O operations during the rebuild (synchronization) process of the failed disk unit may not require additional disk I/O requests. This depends on where the data is read from or written to on the disk unit that is in the synchronization process. For example:

- A read operation from the disk area that already has been rebuilt requires one read operation.
- A read operation from the disk area that has not been rebuilt is treated as a read operation on a failed disk unit.
- A write operation to the disk that has already been rebuilt requires normal read and write operations (two read operations and two write operations).
- A write operation to the disk area that has not been rebuilt is treated as a write operation to a failed disk unit.

Note: The rebuild process takes longer when read and write operations to a replaced disk unit are also occurring. Every read request or every write request interrupts the rebuild process to perform the necessary I/O operations.

4.6.2 Using both device parity protection and mirrored protection

Device parity protection is a hardware function. Auxiliary storage pools and mirrored protection are software functions. When you add disk units and start device parity protection, the disk subsystem or IOP is not aware of any software configuration for the disk units. The software that supports disk protection is aware of which units have device parity protection.

These rules and considerations apply when mixing device parity protection with mirrored protection:

- Device parity protection is not implemented on ASP boundaries.
- Mirrored protection is implemented on ASP boundaries.
- You can start mirrored protection for an ASP even if it currently has no units that are available for mirroring because they all have device parity protection. This ensures that the ASP is always fully protected, even if you add disks without device parity protection later.
- When a disk unit is added to the system configuration, it may be device parity protected.
- For a fully-protected system, you should entirely protect every ASP by device parity protection, by mirrored protection, or both.
- Disk units that are protected by device parity protection can be added to an ASP that has mirrored protection. The disk units that are protected by device parity protection do not participate in mirrored protection (hardware protects them already).
- When you add a disk unit that is not protected by device parity protection to an ASP that has mirrored protection, the new disk unit participates in mirrored protection. Disk units must be added to, and removed from, a mirrored ASP in pairs with equal capacities.
- Before you start device parity protection for disk units that are configured (assigned to an ASP), you must stop mirrored protection for the ASP.
- Before you stop device parity protection, you must stop mirrored protection for any ASPs that contain affected disk units.
- When you stop mirrored protection, one disk unit from each mirrored pair becomes non-configured. You must add the non-configured units to the ASP again before starting mirrored protection.

4.7 Comparing the disk protection options

There are several methods for configuring your system to take advantage of the disk protection features. Before selecting the disk protection options that you want to use, compare the extent of protection that each one provides.

Table 2 provides an overview of the availability tools that can be used on the AS/400 system to protect against different types of failure.

Table 2. Availability tools for the AS/400 system

What is needed	Device parity protection	Mirrored protection	User ASPs
Protection from data loss due to disk-related hardware failure	Yes - See Note 1	Yes	Yes - See Note 4
Maintain availability	Yes	Yes	No
Help with disk unit recovery	Yes - See Note 1	Yes	Yes - See Note 4
Maintain availability when disk controller fails	See Note 3	Yes - See Note 2	No
Maintain availability when disk I/O processor fails	No	Yes - See Note 2	No
Maintain availability when disk I/O bus fails	No	Yes - See Note 2	No
Site disaster protection	No	Yes - See Note 5	No

Notes:

1. Load source unit and any disk units attached to the MFIOP are not protected.
2. Depends on hardware used, configuration, and level of mirrored protection.
3. With device parity protection using the 9337 Disk Array Subsystem, the system becomes unavailable if a controller is lost.
4. With device parity protection using the IOP feature, the system is available as long as the IOP is available.
5. Configuring ASPs can limit the loss of data and the recovery to a single ASP.
6. For site disaster protection, remote mirroring is required.

Be aware of the following considerations when selecting disk protection options:

- With both device parity protection and mirrored protection, the system continues to run after a single disk failure. With mirrored protection, the system may continue to run after the failure of a disk-related component, such as a controller or an IOP.
- When a second disk failure occurs, meaning that the system has two failed disks, the system is more likely to continue to run with mirrored protection than with device parity protection. With device parity protection, the probability of the system failing on the second disk failure can be expressed as *P out of n*. Here, *P* is the total number of disks on the system, and *n* is the number of disks in the device parity set that had the first disk failure. With mirrored protection, the probability of the system failing on the second disk failure is *1 out of n*.

- Device parity protection requires up to 25% additional disk capacity for storage of parity information. The actual increase depends on the number of disk units that are assigned to a device parity set. A system with mirrored protection requires twice as much disk capacity as the same system without mirrored protection. This is because all information is stored twice. Mirrored protection may also require more buses, IOPs, and disk controllers, depending on the level of protection that you want. Therefore, mirrored protection is usually a more expensive solution than device parity protection.
- Usually, neither device parity protection or mirrored protection has a noticeable effect on system performance. In some cases, mirrored protection actually improves system performance. The restore time to disk units protected by device parity protection is slower than the restore time to the same disk devices without device parity protection activated. This is because the parity data must be calculated and written.

4.8 Concurrent maintenance

Concurrent maintenance is the process of repairing or replacing a failed disk-related hardware component while using the system.

Concurrent maintenance allows disks, I/O processors, adapters, power supplies, fans, CD-ROMs, and tapes to be replaced without powering down the server.

On systems without mirrored protection, the system is not available when a disk-related hardware failure occurs. It remains unavailable until the failed hardware is repaired or replaced. However, with mirrored protection, the failing hardware can often be repaired or replaced while the system is being used.

Concurrent maintenance support is a function of system unit hardware packaging. Not all systems support concurrent maintenance.

Mirrored protection only provides concurrent maintenance when it is supported by the system hardware and packaging. The best hardware configuration for mirrored protection also provides for the maximum amount of concurrent maintenance.

It is possible for the system to operate successfully through many failures and repair actions. For example, a failure of a disk head assembly does not prevent the system from operating. A replacement of the head assembly and synchronization of the mirrored unit can occur while the system continues to run. The greater your level of protection, the more often concurrent maintenance can be performed.

On some models, the system restricts the level of protection for Unit 1 and its mirrored unit to *controller-level protection* only. Under some conditions, diagnosis and repair can require active mirrored units to be suspended. You may prefer to power down the system to minimize the exposure of operating with less mirrored protection. Some repair actions require that the system be powered down.

Deferred maintenance is the process of waiting to repair or replace a failed disk-related hardware component until the system can be powered down. The system is available, although mirrored protection is reduced by whatever hardware components have failed. Deferred maintenance is only possible with mirrored protection or device parity protection.

4.9 Redundancy and hot spare

The basic rule for making a server system highly available is to use redundant parts where needed and affordable. Just like the basic idea behind Redundant Array of Independent Disks (RAID), all parts in a server are subject to be a single point of failure. These part can include:

- The CPU
- The power supply
- The main logical board
- The main memory
- Adapter cards

These are all parts that, if even one fails, the overall system is rendered unusable.

To decrease the time involved in replacing a defective component, some customers consider implementing what is known as a *hot spare*. In effect, the customer keeps a local inventory of any component that either:

- Has a higher failure rate than usual
- Has a long lead-time when a replacement is required

Note: The term *hot spare* typically refers to a disk unit. However, the same concept applies to a hot site or another system used for recovery.

Planning for spare disk units

Spare disk units can reduce the time the system runs without mirrored protection after a disk unit failure of a mirrored pair. If a disk unit fails, and a spare unit of the same capacity is available, that spare unit can be used to replace the failed unit. The system logically replaces the failed unit with the selected spare unit. It then synchronizes the new unit with the remaining good unit of the mirrored pair.

Mirrored protection for that pair is again active when synchronization completes (usually less than an hour). However, it may take several hours (from the time a service representative is called until the failed unit is repaired and synchronized) before mirrored protection is again active for that pair.

To make full use of spare units, you need at least one spare unit of each capacity that you have on your system. This provides a spare for any size of disk unit that may fail.

4.10 OptiConnect: Extending a single system

An OptiConnect cluster is a collection of AS/400 systems connected by dedicated fiber optic system bus cables. The systems in an OptiConnect cluster share a common external optical system bus located in an expansion tower or frame. The system providing the shared system bus is called the hub system. Each system that plugs into this shared bus with an OptiConnect Bus Receiver card is called a satellite system. Each satellite system dedicates one of its external system buses that connects to the receiver card in the hub system's expansion tower or rack.

The term OptiConnect link refers to the fiber optic connection between systems in the OptiConnect cluster. The term *path* refers to the logical software established connection between two OptiConnect systems. An OptiConnect network always

consists of at least two AS/400 systems. One of the AS/400 systems is designated as the hub and at least one other system is designated as a satellite.

There are two levels of redundancy available in an OptiConnect cluster:

- **Link redundancy**: Link redundancy is an optical bus hardware feature. Any two systems attached to the hub system shared bus can establish a path between them, including paths to the hub system itself. You can establish path redundancy by configuring two hub systems in the OptiConnect cluster. Each satellite uses two buses to connect with two hub systems. OptiConnect software detects the two logical paths between the two systems and uses both paths for data flow. If a path failure occurs, the remaining path picks up all of the communication traffic.
- **Path redundancy**: The OS/400 infrastructure for any system determines the logical path to another system. It does this by designating which system bus each of the systems that form the path uses. The link between any two satellite systems does not depend on the hub system bus. The two systems use the bus, but the hub system is not involved. Link redundancy is determined by the system models. For OptiConnect clusters, link redundancy is always provided when the extra fiber optic cable is installed. For path redundancy, an extra set of OptiConnect receiver cards and an extra expansion tower or frame are required along with another set of cables.

OptiConnect for OS/400 is an AS/400-to-AS/400 communication clustering solution. It combines unique OptiConnect fiber bus hardware and standard AS/400 bus hardware with unique software. It uses distributed data management (DDM) to allow applications on one AS/400 system to access databases located on other AS/400 systems. The AS/400 systems that contain the databases are the database servers. The remote systems are considered the application client or clients. In most cases, the hub also acts as the database server. Since all systems can communicate with each other (providing that the hub is active), any system can be the client. Some OptiConnect configurations have AS/400 systems that act simultaneously as a server and a client. However, any system can act as a database server.

OptiConnect for OS/400 is a communications vehicle. OptiConnect for OS/400 products provide AS/400 systems with physical links for a high availability clustered solution. OptiConnect for OS/400 components support the infrastructure for applications to conduct data exchanges over high speed connections.

OptiConnect for OS/400 does not offer high availability with applications that utilize the hardware links. OptiConnect for OS/400 can be the transport mechanism for in-house developed applications, business partner software, or remote journal support.

Further information on OptiConnect is found in 6.8, “Bus level interconnection” on page 82, and 6.8.1, “Bus level interconnection and a high availability solution” on page 84.

4.11 Cluster support

For planned or unplanned outages, clustering and system mirroring offer the most effective solution. For customers requiring better than 99.9% system availability, AS/400 clusters are viable.

Cluster solutions connect multiple AS/400 systems together with various interconnect fabrics, including high-speed optical fiber, to offer a solution that can deliver up to 99.99% system availability.

High availability is achieved with an alternative system that replicates the availability of the production system. These systems are connected by high-speed communications and use replication software to achieve this. They also require enough DASD to replicate the whole or critical part of the production system. With the entire system replicated, the mirrored system can enable more than just a disaster recovery solution.

Combining these clusters with software from AS/400 high-availability business partners (such as those described in Chapter 10, “High availability business partner solutions” on page 111) improves the availability of a single AS/400 system by replicating business data to one or more AS/400 systems. This combination can provide a disaster recovery solution.

Clusters are a configuration or a group of independent servers that appear on a network as a single machine. As illustrated in Figure 11, a cluster is a collection of complete systems that work together to provide a single and unified computing resource

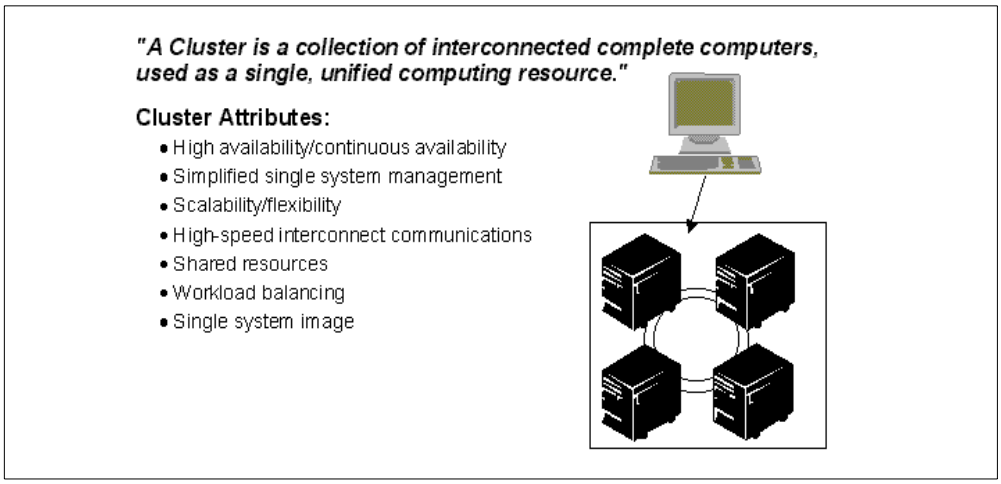


Figure 11. Cluster definition

This cluster group is managed as a single system or operating entity and is designed specifically to tolerate component failures and to support the addition or subtraction of components in a way that is transparent to users. Clusters allow you to efficiently group systems together to set up an environment that provides availability that approaches 100% for critical applications and critical data. Resources can be accessed without regard to location. A client interacts with a cluster as if it were a single system.

With the introduction of clusters, the AS/400e system offers a continuous availability solution if your business demands operational systems 24 hours a day, 365 days a year (24 x 365). This solution, called OS/400 Cluster Resource Services, is part of the OS/400 operating system. It provides failover and switchover capabilities for your systems that are used as database servers or application servers. If a system outage or a site loss occurs, the functions that are provided on a clusters server system can be switched over to one or more designated backup systems that contain a current copy (replica) of your critical resource. The failover can be automatic if a system failure should happen, or if you can control how and when the transfer takes place by manually initiating a switchover.

Cluster management tools control the cluster from anywhere in the network. End users work on servers in the cluster without knowing or caring where their applications are running.

In the event of a failure, Cluster Resource Services (CRS), which is running on all systems, provides a switchover. This switch causes minimal impact to the end user or applications that are running on a server system. Data requesters are automatically rerouted to the new primary system. You can easily maintain multiple data replications of the same data.

Any AS/400 model that can run OS/400 V4R4 or later is compatible for implementing clustering. You must configure Transmission Control Protocol/Internet Protocol (TCP/IP) on your AS/400e systems before you can implement clustering. In addition, you can purchase a cluster management package from a High Availability Business Partner (HAV BP) that provides the required replication functions and cluster management capabilities.

Refer to *AS/400 Clusters: A Guide to Achieving Higher Availability*, SG24-5194, for further information.

4.12 LPAR hardware perspective

Logical partitions allow you to run multiple independent OS/400 instances or partitions. Figure 12 shows a basic LPAR configuration. For V4R5, each partition has has its own processors, memory, and disks. For V5R1, resources can be share between partitions. With logical partitioning, you can address multiple system requirements in a single machine to achieve server consolidation, business unit consolidation, and mixed production and test environments. You can run a cluster environment on a single system image. LPAR support is available on n-way symmetric multiprocessing iSeries models 8xx and AS/400 models 6xx, Sxx, and 7xx. See 7.5, “Logical Partition (LPAR) support” on page 93.

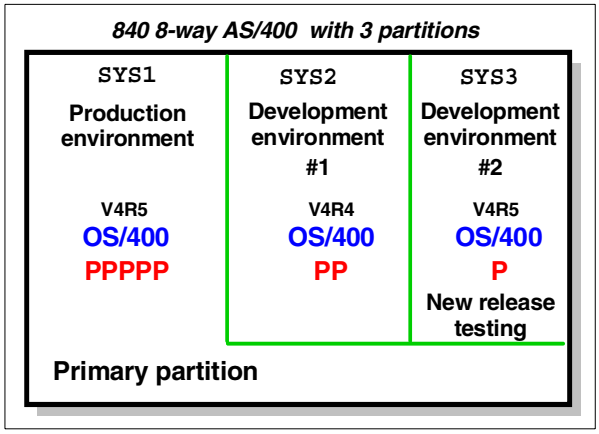


Figure 12. A basic LPAR configuration

By itself, LPAR does not provide a significant availability increase. It can, however, be used to complement other availability strategies.

See 7.5, “Logical Partition (LPAR) support” on page 93, for a discussion of LPAR from an OS/400 view.

4.12.1 Clustering with LPAR support

Since each partition on an LPAR system is treated as a separate server, you can run a cluster environment on a single system image. One cluster node per CPU can exist within one LPAR system.

Clustering partitions can provide for a more cost efficient clustering solution than multiple systems. However, an LPAR clustered environment increases single points of failure. For example, if the server’s primary partition becomes unavailable, all secondary partitions also become unavailable (the opposite is not true).

In some environments, LPAR ideally lends itself to situations where both a local and remote backup server is desired. A good example is when a business works to provide its own disaster recovery capability.

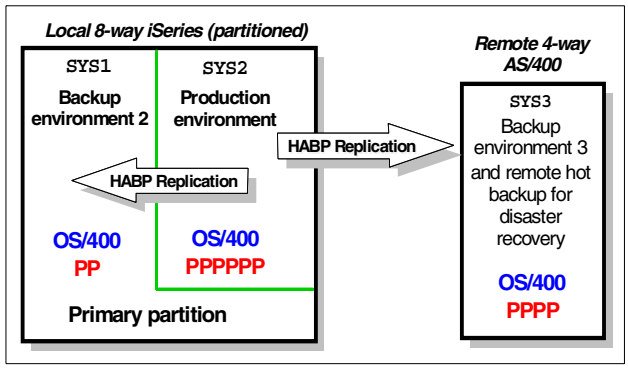


Figure 13. LPAR, local and remote iSeries and AS/400 cluster

The highest level of availability is obtained with two separate servers. Figure 13 shows that, with clustering active, data is replicated to both the local backup server and the remote server. In the event of a disaster (or the need for the entire local hardware to be powered off), the remote backup server is available. In some cases, this is more cost-efficient (including floor space) than separate servers.

Integrated availability options

In most cases, it is recommended that integrated availability solutions be used with a cluster to further mask or reduce downtime and to increase a cluster's efficiency. Consider the following list:

- Disk protection: *Device Parity Protection* (RAID-5) and OS/400 *Disk Mirroring*.
- Auxiliary storage pools (ASPs)
- Access path protection
- Logical Partitions (LPAR)

In all cases, it is highly recommended that these integrated availability options be used in a clustered environment, as well as on a standalone iSeries or AS/400 server.

4.13 UPS

An uninterruptible power supply (UPS) provides auxiliary power to the processing unit, disk units, the system console, and other devices that you choose to protect from power loss. When you use a UPS with the AS/400 system, you can:

- Continue operations during brief power interruptions (brown outs).
- Protect the system from voltage peaks (white outs).
- Provide a normal end of operations that reduces recovery time when the system is restarted. If the system abnormally ends before completing a normal end of operations, recovery time is significant.

Normally, a UPS does not provide power to all local workstations. The UPS also usually does not provide power to modems, bridges, or routers that support remote workstations. Consider supplying alternate power to both workgroups since the inability of worker access to information disrupts productivity. You can avoid such disruption with proper availability and recovery implementation.

Also, design your interactive applications to handle the loss of communication with a workstation. Otherwise, system resources are used in an attempt to recover devices that have lost power. Refer to Chapter 12, “Communications Error Recovery and Availability” in *The System Administrator’s Companion to AS/400 Availability and Recovery*, SG24-2161, for more information on resources used during device recovery.

The programming language reference manuals provide examples of how to use the error feedback areas to handle workstations that are no longer communicating with the application. *Backup and Recovery*, SC41-5304, describes how to develop programs to handle an orderly shutdown of the system when the UPS takes over.

4.14 Battery backup

Most (but not all) AS/400 models are equipped with a battery backup. Based on the system storage size, relying on a battery backup for enough time for an orderly shutdown is not sufficient.

The battery capacity typically varies between 10 and 60 minutes. The useful capacity depends on the application requirements, main storage size, and system configuration. Consider the reduction of capacity caused by the natural aging of the battery and environmental extremes of the site when selecting the battery. The battery must have the capacity to maintain the system load requirements at the end of its useful life.

Refer to *Backup and Recovery*, SC41-5304, for power down times for the advanced series systems. Refer to the *AS/400 Physical Planning Reference*, SA41-5109, for power down times for the AS/400 Bnn-Fnn models. Also, refer to the *Physical Planning Reference* for later AS/400 models at the Web site:
`http://www.as400.ibm.com/tstudio/planning/index.rf.htm`

4.15 Continuously powered main storage

On V3R6 systems and later, AS/400 systems are equipped with a System Power Control Network (SPCN) feature. This provides the Continuously Powered Main Storage (CPM) function. During a power fluctuation, the transition to CPM mode is 90 seconds after an initial 30 second waiting period. The internal battery backup provides sufficient power to keep the AS/400 system up for the 120 seconds until the transition to the CPM is complete. With CPM enabled, the battery provides sufficient power to shut down the system and maintain the contents of memory for up to 48 hours after a power loss without user interface or control.

The transition to CPM is irreversible. CPM interrupts the processes at the next microcode end statement and forces as many updates to the disk as it can. During the next IPL, it restores main storage and attempts to complete outstanding updates. Preserving main storage contents significantly reduces the amount of time the system requires to perform an IPL after a power loss. CPM operates outside of transaction boundaries.

You can use the CPM feature along with a UPS (or the battery backup). If the system detects that the UPS can no longer provide sufficient power to the system, the data currently in memory is put into “sleep” mode. The CPM storage feature takes control and maintains data in memory for up to 48 hours. With the CPM feature, the system automatically initiates an IPL after power is restored.

CPM is a viable feature. Choosing to use CPM depends on your expectations of your local power and battery backup or generator to maintain power at all times.

Refer to *Backup and Recovery*, SC41-5304, for more information on CPM requirements.

4.16 Tape devices

For information on what tape devices are available for each AS/400 model, and the hardware and software requirements to support each model, refer to the *iSeries Handbook*, GA19-5486, and *iSeries and AS/400e System Builder*, SG24-2155.

For save and restore performance rates, see Appendix C, “Save and Restore Rates of IBM Tape Drives for Sample Workloads”, and Section 8.1, “Save and Restore Performance” in the *AS/400 Performance Capabilities Manual* at:
`http://publib.boulder.ibm.com/pubs/pdfs/as400/V4R5PDF/AS4PPCP3.PDF`

4.16.1 Alternate installation device

On V4R1 (and later) systems, you can use a combination of devices that are attached on the first system bus, as well as additional buses. The alternate installation device does not need to be attached to the first system bus. For example, the 3590 tape drive can be positioned up to 500 meters or two kilometers away. This enables a physical security improvement since users who are allowed access to the machine room may be different than those operating the tape drives.

You can select an alternate installation device connected through any I/O bus attached to the system. When you perform a D-mode IPL (D-IPL), you can use the tape device from another bus using the Install Licensed Internal Code display. For example, if you have a 3590 attached to another bus (other than Bus 1), you can choose to install from the alternate installation device using the Install Licensed Internal Code display and then continue to load the LIC, OS/400, and user data using the alternate installation device.

Note: Set up alternate installation device support prior to performing a D-IPL. System Licensed Internal Code (SLIC) media is necessary to perform the D-IPL that restores and installs from the tape device.

Recommendation

Before using the alternate installation device, ensure that it is defined on a bus other than system Bus 1. You must enable the device. When installing from the alternate installation device, you need both your tape media and the CD-ROM media containing the Licensed Internal Code.

Some models (typically with 3590 tape devices attached) experience a performance improvement when using an alternate installation device for other save and restore or installation operations. This is caused by having the tape drive on a different IOP than the one to which the load source unit is attached. On systems prior to V4R1, the alternate installation device is only supported using devices attached to the first system bus. The first system bus connects to the service processor IOP. Typically, this is where the optical or tape devices used for installations are attached.

Chapter 5. Auxiliary storage pools (ASPs)

An auxiliary storage pool (ASP) is a software definition of a group of disk units on your system. This means that an ASP does not necessarily correspond to the physical arrangement of disks. Conceptually, each ASP on your system is a separate pool of disk units for single-level storage. The system spreads data across the disk units within an ASP. If a disk failure occurs, you need to recover only the data in the ASP that contained the failed unit. Prior to V5R1, here are two types of ASPs:

- System auxiliary storage pool
- User auxiliary storage pools

Note

Independent ASPs are introduced at V5R1. At the time this Redpaper was written, the appropriate information was not available.

Your system may have many disk units attached to it that are optionally assigned to an auxiliary storage pool. To your system, the pool looks like a single unit of storage. The system spreads objects across all disk units. You can use auxiliary storage pools to separate your disk units into logical subsets.

When you assign the disk units on your system to more than one ASP, each ASP can have different strategies for availability, backup and recovery, and performance. ASPs provide a recovery advantage if the system experiences a disk unit failure resulting in data loss. If this occurs, recovery is only required for the objects in the ASP that contained the failed disk unit. System objects and user objects in other ASPs are protected from the disk failure. There are also additional benefits and certain costs and limitations that are inherent in using ASPs.

5.1 Deciding which ASPs to protect

Because mirrored protection is configured by auxiliary storage pool, the ASP is the user's level of control over single-level storage. Mirrored protection can be used to protect one, some, or all ASPs on a system. However, multiple ASPs are not required in order to use mirrored protection.

Mirrored protection works well when all disk units on a system are configured into a single ASP (the default on the AS/400 system). In fact, mirroring reduces the need to partition auxiliary storage into ASPs for data protection and recovery. However, ASPs may still be recommended for performance and other reasons.

To provide the best protection and availability for the entire system, mirror all ASPs in the system. Consider the following situations:

- If the system has a mixture of some ASPs with and without mirrored protection, a disk unit failure in an ASP without mirrored protection severely limits the operation of the entire system. Data can be lost in the ASP in which the failure occurred. A long recovery may be required.

- If a disk fails in a mirrored ASP, and the system also contains ASPs that are not mirrored, data is not lost. However, in some cases, concurrent maintenance may not be possible.

The disk units that are used in user ASPs should be selected carefully. For best protection and performance, an ASP should contain disk units that are attached to several different I/O processors. The number of disk units in the ASP that are attached to each I/O processor should be the same (that is, balanced).

ASPs are further discussed in Chapter 5, “Auxiliary storage pools (ASPs)” on page 63.

5.1.1 Determining the disk units needed

A mirrored ASP requires twice as much auxiliary storage as an ASP that is not mirrored. This is because the system keeps two copies of all the data in the ASP. Also, mirrored protection requires an even number of disk units of the same capacity so that disk units can be made into mirrored pairs.

On an existing system, note that it is not necessary to add the same types of disk units already attached to provide the required additional storage capacity. Any new disk units may be added as long as sufficient total storage capacity and an even number of storage units of each size are present. The system assigns mirrored pairs and automatically moves the data as necessary. If an ASP does not contain sufficient storage capacity, or if storage units cannot be paired, mirrored protection cannot be started for that ASP.

The process of determining the disk units needed for mirrored protection is similar for existing or new systems. Review the following points to plan disk requirements:

1. Determine how much data each ASP contains.
2. Determine a target percent of storage used for the ASP (how full the ASP will be).
3. Plan the number and type of disk units needed to provide the required storage. For an existing ASP, you can plan a different type and model of disk unit to provide the required storage.

After planning for all ASPs is complete, plan for spare units, if desired. Once you know all of this information, you can calculate your total storage needs.

The planned amount of data and the planned percent of storage used work together to determine the amount of actual auxiliary storage needed for a mirrored ASP. For example, if an ASP contains 1 GB (GB equals 1,073,741,824 bytes) of actual data, it requires 2 GB of storage for the mirrored copies of the data. If 50% capacity is planned for that ASP, the ASP needs 4 GB of actual storage. If the planned percent of storage used is 66%, 3 GB of actual storage is required. One gigabyte of real data (2 GB of mirrored data) in a 5 GB ASP results in a 40% auxiliary storage utilization.

Total planned storage capacity needs

After planning for the number and type of storage units needed for each ASP on the system, and for any spare storage units, add the total number of storage units of each disk unit type and model.

The number planned is the number of storage units of each disk unit type, not the number of disk units.

The following section provides a more detailed description.

5.2 Assigning disk units to ASPs

If you decide that you want more than one auxiliary storage pool (ASP), make the following determinations for each ASP:

- How much storage do you need?
- What disk protection (if any) should you use?
- Which disk units should be assigned?
- Which objects should be placed in the ASP?

The *Workstation Customization Programming* book*,* SC41-5605, provides information to help you with these considerations. This book is only available online at the AS/400 Library at: `http://as400bks.rochester.ibm.com`

At the site, click **AS/400 Information Center**. Select your language and click **GO!** Click **V4R4** and then click **Search or view all V4R4 books**. Enter the book number in the seach field and click **Find**. Finally, click the appropriate publication that appears.

When you work with disk configuration, you may find it helpful to begin by making a list of all the disks and disk-related components on your system. You can put this information in a chart like Table 3, or you may want to draw a diagram.

Table 3. Disk configuration example chart

IOP	Controller	Unit	Type and model	Type and model	Capacity	Resource name	Name of mirrored pair
1	00	01	1	6602-030	1031	1	DD001
1	10	01	2	6602-074	733	1	DD019
1	10	02	3	6602-070	1031	1	DD036
1	00	02	6	6602-030	1031	1	DD002
1	10	03	4	6602-074	773	3	DD005
1	10	04	5	6602-074	773	3	DD033

5.3 Using ASPs

User ASPs are used to manage the following system performance and availability requirements:

- Provide dedicated resources for frequently used availability objects, such as journal receivers
- Allow online and unattended saves.
- Place infrequently used objects, such as large history files, on disk units with slower performance

5.3.1 Using ASPs for availability

Different parts of your system may have different requirements for availability and recovery. For example, you may have a large history file that is changed only at the end of the month. The information in the file is useful but not critical. You may put this file in a separate library in a user ASP that does not have any disk protection (mirrored protection or device parity protection). You could omit this library from your daily save operations and choose to save it only at the end of the month when it is updated.

Another example would be documents and folders. Some documents and folders are critical to the organization and should be protected with device parity protection or mirrored protection. They can be put in a protected user ASP. Others are kept on the system to provide information but do not change very often. They can be in a different user ASP with a different strategy for saving and for protection.

5.3.2 Using ASPs to dedicate resources or improve performance

If you are using user ASPs for better system performance, consider dedicating the ASP to one object that is very active. In this case, you can configure the ASP with only one disk unit. However, it usually does not improve performance to place a single device-parity protected unit in a user ASP because the performance of that unit is affected by other disk units in the device parity set.

Refer to Figure 14 for a visual example of multiple ASPs.

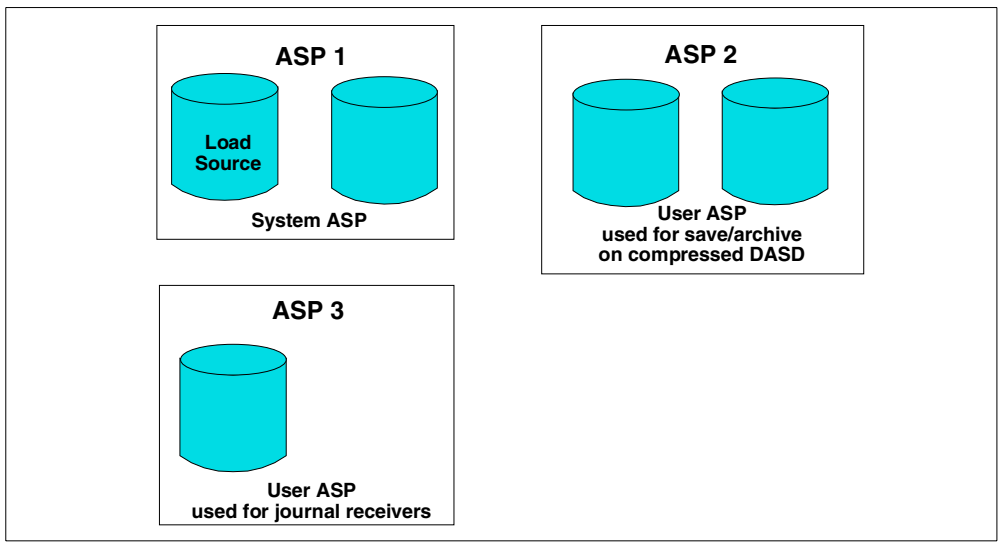


Figure 14. Auxiliary storage pools

Allocating one user ASP exclusively for journal receivers that are attached to the same journal can improve journaling performance. By having the journal and database files in a separate ASP from the attached receivers, there is no contention for journal receiver write operations. The units that are associated with the ASP do not have to be repositioned before each read or write operation. Journaling uses as many as 10 disk arms when writing to a journal receiver. Configuring an ASP with more than 10 arms does not provide any additional performance advantage for journaling. However, if you do have an ASP with more than 10 arms, journaling uses the 10 fastest arms. If you add more disk units to

the ASP while the system is active, the system determines whether to use the new disk units for journal receivers the next time the change journal function is performed.

Another method for improving performance is to make sure that there are enough storage units in the user ASP to support the number of physical input and output operations that are done against the objects in the user ASP. You may have to experiment by moving objects to a different user ASP and then monitor performance in the user ASP to see if the storage units are used excessively. If the units show excessive use, you should consider adding more disk units to the user ASP.

5.3.3 Using ASPs with document library objects

You can place document library objects (DLOs) in user ASPs. The possible advantages of placing DLOs in user ASPs are:

- The ability to reduce save times for DLOs and to separate them by their save requirements.
- The ability to separate DLOs by availability requirements. Critical DLOs can be placed in user ASPs that are protected by mirrored protection or device parity protection. DLOs that change infrequently can be placed in unprotected ASPs with slower drives.
- The ability to grow to a larger number of documents. If you have V3R7 or a later release of the OS/400 licensed program, you can run multiple SAVDLO or RSTDLO procedures against different ASPs. If you have V4R1 or a later release of the OS/400 licensed program, you can run multiple SAVDLO operations on the same ASP.

One approach for placing DLOs in user ASPs is to leave only system DLOs (IBM-supplied folders) in the system ASP. Move other folders to user ASPs. The system folders do not change frequently, so they can be saved infrequently.

You can specify an ASP on the SAVDLO command. This allows you to save all the DLOs from a particular ASP on a given day of the week. For example, you could save DLOs from ASP 2 on Monday, DLOs from ASP 3 on Tuesday, and so on. You could save all changed DLOs daily.

The recovery steps if you use this type of save technique would depend on what information was lost. If you lost an entire ASP, you would restore the last complete saved copy of DLOs from that ASP. You would then restore the changed DLOs from the daily saves.

When you save DLOs from more than one ASP in the same operation, a different file and a sequence number are created on the tape for each ASP. When you restore, you must specify the correct sequence number. This makes it simple to restore the changed DLOs only to the ASP that was lost without needing to know all the folder names.

These restrictions and limitations apply when placing DLOs in user ASPs:

- When using a save file for a save operation, you can save DLOs from only one ASP.
- When using an optical file for a save operation, you can save DLOs from only one ASP.

- If you are saving to a save file and you specify SAVDLO DLO(*SEARCH) or SAVDLO DLO(*CHG), you must also specify an ASP, even if you know the results of you search are found in a single ASP.
- Documents that are not in folders must be in the system ASP.
- Mail can be filed into a folder on a user ASP. Unfiled mail is in the system ASP.

Note: When you specify DLO(*SEARCH) or DLO(*CHG) for the SAVDLO command, specify an ASP, if possible. Specifying an ASP saves system resources.

5.3.4 Using ASPs with extensive journaling

If journals and files being journaled are in the same ASP as the receivers and the ASP overflows, you must end journaling of all files and recover from the overflowed condition for the ASP. Backup and Recovery describes how to recover an overflowed ASP.

If the journal receiver is in a different ASP than the journal, and the user ASP that the receiver is in overflows, perform the following steps:

1. Create a new receiver in a different user ASP.
2. Change the journal (CHGJRN command).
3. Save the detached receiver.
4. Delete the detached receiver.
5. Clear the overflowed ASP without ending journaling.
6. Create a new receiver in the cleared ASP.
7. Attach the new receiver with the CHGJRN command.

5.3.5 Using ASPs with access path journaling

If you plan to use explicit access path journaling, IBM recommends that you first change the journal to a journal receiver in the system ASP (ASP 1) for a few days. Start access path journaling to see storage requirements for the receiver before you allocate the specific size for a user ASP.

5.3.6 Creating a new ASP on an active system

Beginning with V3R6 of the OS/400 licensed program, you can add disk units while your system is active. When you add disk units to an ASP that does not currently exist, the system creates a new ASP. If you choose to create a new user ASP while your system is active, be sure you understand the following considerations:

- You cannot start mirrored protection while the system is active. The new ASP is not fully protected unless all of the disk units have device parity protection.
- You cannot move existing disk units to the new ASP while your system is active.The system must move data when it moves disk units. This can be done only through Dedicated Service Tools (DST).
- The system uses the size of an ASP to determine the storage threshold for the journal receivers that are used by system-managed access-path protection (SMAPP).

  When you create an ASP while your system is active, the size of the disk units that you specify on the operation that creates the ASP is considered the size of the ASP for SMAPP. For example, assume that you add two disk units to a

new ASP (ASP 2). The total capacity of the two disk units is 2,062 MB. Later, you add two more disk units to increase the capacity to 4,124 MB. For the purposes of SMAPP, the size of the ASP remains 2,062 MB until the next time you perform an IPL. This means that the storage threshold of your SMAPP receivers is lower and the system must change receivers more often. Usually, this does not have a significant impact on system performance.

The system determines the capacity of every ASP when you perform an IPL. At that time, the system makes adjustments to its calculations for SMAPP size requirements.

5.3.7 Making sure that your system has enough working space

When you make changes to your disk configuration, the system may need working space. This is particularly true if you plan to move disk units from one ASP to another. The system needs to move all the data from the disk unit to other disk units before you move it. There are system limits for the amount of auxiliary storage.

If your system does not have sufficient interim storage, begin by cleaning up your disk storage. Many times, users keep objects on the system, such as old spooled files or documents, when these objects are no longer needed. Consider using the automatic cleanup function of Operational Assistant to free some disk space on your system.

If cleaning up unnecessary objects in auxiliary storage still does not provide sufficient interim disk space, another alternative is to remove objects from your system temporarily. For example, if you plan to move a large library to a new user ASP, you can save the library and remove it from the system. You can then restore the library after you have moved disk units. Here is an example of how to accomplish this:

1. Save private authorities for the objects on your system by typing `SAVSECDTA DEV` (*tape-device*).
2. Save the object by using the appropriate SAVxxx command. For example, to save a library, use the SAVLIB command. Consider saving the object twice to two different tapes.
3. Delete the object from the system by using the appropriate DLTxxx command. For example, to delete a library, use the DLTLIB command.
4. Recalculate your disk capacity to determine whether you have made sufficient interim space available.
5. If you have enough space, perform the disk configuration operations.
6. Restore the objects that you deleted.

5.3.8 Auxiliary storage pools: Example uses

The following list explains how ASPs are used to manage system performance and backup requirements:

- You can create an ASP to provide dedicated resources for frequently used objects, such as journal receivers.

- You can create an ASP to hold save files. Objects can be backed up to save files in a different ASP. It is unlikely that both the ASP that contains the object and the ASP that contains the save file will be lost.
- You can create different ASPs for objects with different recovery and availability requirements. For example, you can put critical database files or documents in an ASP that has mirrored protection or device parity protection.
- You can create an ASP to place infrequently used objects, such as large history files, on disk units with slower performance.
- You can use ASPs to manage recovery times for access paths for critical and noncritical database files using system-managed access-path protection.

5.3.9 Auxiliary storage pools: Benefits

Placing objects in user ASPs can provide several advantages, including:

- **Additional data protection**: By separating libraries, documents, or other objects in a user ASP, you protect them from data loss when a disk unit in the system ASP or other user ASPs fails. For example, if you have a disk unit failure, and data contained on the system ASP is lost, objects contained in user ASPs are not affected and can be used to recover objects in the system ASP. Conversely, if a failure causes data that is contained in a user ASP to be lost, data in the system ASP is not affected.
- **Improved system performance**: Using ASPs can also improve system performance. This is because the system dedicates the disk units that are associated with an ASP to the objects in that ASP. For example, suppose you are working in an extensive journaling environment. Placing libraries and objects in a user ASP can reduce contention between the journal receivers and files if they are in different ASPs. This improves journaling performance. However, placing many active journal receivers in the same user ASP is not productive. The resulting contention between writing to more than one receiver in the ASP can slow system performance. For maximum performance, place each active journal receiver in a separate user ASP.
- **Separation of objects with different availability and recovery requirements**: You can use different disk protection techniques for different ASPs. You can also specify different target times for recovering access paths. You can assign critical or highly used objects to protected high-performance disk units. You may assign large low-usage files, like history files, to unprotected low-performance disk units.

5.3.10 Auxiliary storage pools: Costs and limitations

There are some specific limitations that you may encounter when using auxiliary storage pools (ASPs):

- The system cannot directly recover lost data from a disk unit media failure. This situation requires you to perform recovery operations.
- Using ASPs can require additional disk devices.
- Using ASPs requires you to manage the amount of data in an ASP and avoid an overflowed ASP.
- You need to perform special recovery steps if an ASP overflows.
- Using ASPs requires you to manage related objects. Some related objects, such as journals and journaled files, must be in the same ASP.

5.4 System ASP

The system automatically creates the system ASP (ASP 1). This contains disk Unit 1 and all other configured disks that are not assigned to a user ASP. The system ASP contains all system objects for the OS/400 licensed program and all user objects that are not assigned to a user ASP.

Note: You can have disk units that are attached to your system but are not configured and are not being used. These are called non-configured disk units. There are additional considerations that you should be aware of regarding the capacity of the system ASP and protecting your system ASP. These are explained in the following sections.

5.4.1 Capacity of the system ASP

If the system ASP fills to capacity, the system ends abnormally. If this occurs, you must perform an IPL of the system and take corrective action (such as deleting objects) to prevent this from re-occurring. You can also specify a threshold that, when reached, warns the system operator of a potential shortage of space. For example, if you set the threshold value at 80 for the system ASP, the system operator message queue (QSYSOPR) and the system message queue (QSYSMSG) are notified when the system ASP is 80% full. A message is sent every hour until the threshold value is changed, or until objects are deleted or transferred out of the system ASP. If you ignore this message, the system ASP fills to capacity and the system ends abnormally.

A third method for preventing the system ASP from filling to capacity is to use the QSTGLOWLMT and QSTGLOWACN system values.

5.4.2 Protecting your system ASP

IBM recommends that you use device parity protection or mirrored protection on the system ASP. Using disk protection tools reduces the chance that the system ASP will lose all data. If the system ASP is lost, addressability to objects in every user ASP is also lost.

You can restore the addressability by restoring the entire system or by running the Reclaim Storage (RCLSTG) command. However, the RCLSTG command cannot recover object ownership. After you run the command, the QDFTOWN user profile owns all objects found without ownership intact. You can use the Reclaim Document Library Object (RCLDLO) command procedure to recover ownership of document library objects.

5.5 User ASPs

Grouping a set of disk units together and assigning that group to an auxiliary storage pool (ASP) creates a user ASP. You can configure user ASPs 2 through 16. They can contain libraries, documents, and certain types of objects. There are two types of user ASPs:

- History file
- Non-library user ASPs

Once you have ASPs configured, you should protect them by using mirroring or device parity protection.

5.5.1 Library user ASPs

Library user ASPs contain libraries and document library objects (DLOs). It is recommended that you use library user ASPs because the recovery steps are easier than with non-library user ASPs. You should be familiar with the following regulations regarding library user ASPs:

- Do not create system or product libraries (libraries that begin with a Q or #)or folders (folders that begin with a Q) in a user ASP. Do not restore any of these libraries or folders to a user ASP. Doing so can cause unpredictable results.
- Library user ASPs may contain both libraries and document library objects. The document library for a user ASP is called QDOCnnnn (here, *nnnn* is the number of the ASP).
- Journals and files that are being journaled must be in the same ASP. Place the journal receivers in a different ASP. This protects against the loss of the files and the receivers if a disk media failure occurs.
- Journaling cannot be started on an object (STRJRNPF or STRJRNAP command) if the journal (object type *JRN) and the object to be journaled are in different ASPs.
- Journaling cannot be started again for a file that is saved and then restored to a different ASP that does not contain the journal. The journal and the file must be in the same ASP for journaling to be automatically started again for the file.
- No database network can cross ASP boundaries.
- You cannot create a file in one ASP that depends on a file in a different ASP. All based-on physical files for a logical file must be in the same ASP as the logical file. The system builds access paths only for database files in the same ASP as the based-on physical file (temporary queries are not limited). Access paths are never shared by files in different ASPs. Record formats are not shared between different ASPs. Instead, a format request is ignored and a new record format is created.
- You can place an SQL collection in a user ASP. You specify the target ASP when you create the collection.
- If the library user ASP does not contain any database files, set the target recovery time for the ASP to *NONE. This would be true, for example, if the library user ASP contains only libraries for journal receivers. If you set the access path recovery time to *NONE, this prevents the system from doing unnecessary work for that ASP. The *Backup and Recovery Guide*, SC41-5304, describes how to set access path recovery times.

Non-library user ASPs

Non-library user ASPs contain journals, journal receivers, and save files with libraries that are in the system ASP. If you are assigning access path recovery times for individual ASPs, you should set the target recovery time for a non-library user ASP to *NONE. A non-library user ASP cannot contain any database files and cannot, therefore, benefit from SMAPP.

If you set an access path recovery time for a non-library user ASP to a value other than *NONE, the system performs extra work with no possible benefit.

Backup and Recovery Guide, SC41-5304, describes how to set access path recovery times.

Using ASPs can require protecting user ASPs. Keep the following points in mind regarding user ASP protection:

- All ASPs, including the system ASP, should have mirrored protection or consist entirely of disk units with device parity protection to ensure that the system continues to run after a disk failure in an ASP.
- If a disk failure occurs in an ASP that does not have mirrored protection, the system may not continue to run. This depends on the type of disk unit and the error.
- If a disk failure occurs in an ASP that has mirrored protection, the system continues to run (unless the both storage units of a mirrored have failed).
- If a disk unit fails in an ASP that has device parity protection, the system continues running as long as no other disk unit in the same device parity set fails.
- System limits are set for auxiliary storage. During an IPL, the system determines how much auxiliary storage is configured on the system. The total amount is the sum of the capacity of the configured units and their mirrored pairs (if any). Disk units that are not configured are not included. The amount of disk storage is compared to the maximum that is supported for a particular model.
- If more than the recommended amount of auxiliary storage is configured, a message (CPI1158) is sent to the system operator's message queue (QSYSOPR) and the QSYSMSG message queue (if it exists on the system). This message indicates that too much auxiliary storage exists on the system. This message is sent once during each IPL as long as the amount of auxiliary storage on the system is more than the maximum amount supported.

Chapter 6. Networking and high availability

One of the major items to consider for availability is the network. When planning for a network, capacity and accessibility are addressed, just as capacity and accessibility are planned for the system itself.

Your company needs a stable computing environment to fuel its business growth and to sharpen its advantage in a competitive marketplace. When major hubs or routers are down, users have difficulty accessing key business applications. The ability to carry on business, such as enrolling new members to a bank, or to respond to member inquiries, is impacted. To minimize this effect, aim for stringent metrics (for example, 99.5 percent availability of the operational systems).

Ultimately, the network must be sound and recoverable to support core business applications. Employ a carefully planned comprehensive network management solution that is:

- Scalable and flexible, no matter how complicated the task
- Provides around-the-clock availability and reliability of applications to users, no matter where they’re located
- Capable of building a solid network foundation, no matter how complex the system

This chapter comments on the various network components of an HA solution and how they can affect the overall availability.

6.1 Network management

Networks typically comprise a wide variety of devices, such as hubs, routers, bridges, switches, workstations, PCs, laptops, and printers. The more different components and protocols there are in a network, the more difficult it is to manage them.

Problem detection and response can be at a local or remote level. Tools can help you correct problems at the source before users are affected and minimize downtime and performance problems.

Management tools provide a detailed view of the overall health and behavior of your network's individual elements. These elements, such as routers, servers, end systems, data, event flow, and event traffic are useful to record. When network problems occur, use management tools to quickly identify and focus on the root cause of the error.

The focal point of control is where all network information is viewed to provide a complete picture of what is happening. This console or server provides monitoring and alert actions, as well as maintenance capabilities.

With proper network management, the time to respond to and resolve a network error is reduced.

6.2 Redundancy

Availability is measured as the percentage of time that online services for a critical mass of end users can function at the end-user level during a customer's specified online window.

When systems are combined into a cluster, the supporting network must be able to maintain the same level of availability as the cluster nodes. Communication providers must deliver guarantees that they will be available for, and have sufficient capacity for, all possible switch scenarios. There must be alternate network paths to enable the cluster services to manage the cluster resources.

Redundant paths can prevent a cluster partition outage from occurring. Redundant network connections are illustrated in Figure 15.

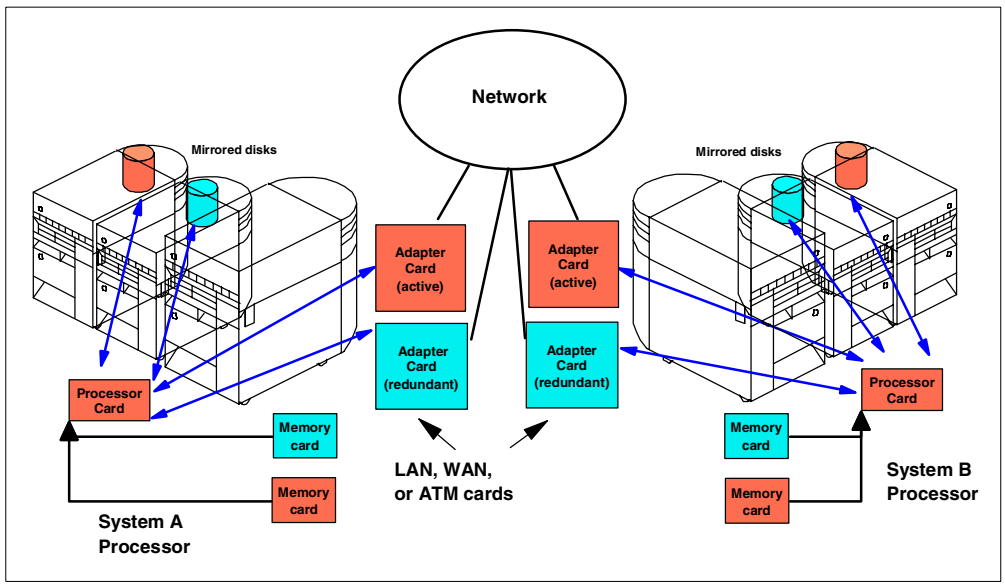


Figure 15. Redundant network connections

As shown in Figure 15, the adapter card can be a LAN, WAN, or ATM adapter. Duplicates are installed, as well as duplicate memory and disk units. Indeed, System A serves as a duplicate system to System B, and vice versa.

When a network connection between two machines that are part of the business process does not function, it does not necessarily mean the business process is in danger. If there is an alternative path and, in case of malfunctioning, the alternative path is taken automatically, monitoring and handling these early events assures a minimum breakage.

6.3 Network components

Protocol, physical connection, the hardware and software needed, the backup options available, and network design are all factors to consider for a highly available network.

Components involved in a network include:

- **Protocols**:
  - **Bisynchronous**
  - **Asynchronous**
  - **Systems Network Architecture (SNA)**: The predominate protocol for connecting terminals to host systems. SNA has strong support for congestion control, flow control, and traffic prioritization. It is known to provide stable support for large and complex networks. However, SNA is not capable of being routed natively across a routed network. Therefore, it must be encapsulated to flow across such a network.
  - **Transmission Control Protocol/Internet Protocol (TCP/IP)**: TCP/IP is the protocol used to build the world's largest network, the Internet. Unlike SNA, all hosts are equal in TCP/IP. The protocol can be carried natively through a routing network. TCP/IP is the de facto published standard. This allows many vendors to interoperate between different machines. Poor congestion control, flow control, traffic prioritization, and a lack of controls makes it difficult to guarantee a response time over a wide area network.
  - **Internetwork Packet Exchange (IPX) was developed for NetWare**: NetWare is a network operating system and related support services environment introduced by Novell, Inc. It is composed of several different communication protocols. Services are provided in a client/server environment in which a workstation (client) requests and receives the services that are provided by various servers on the network. Since this is usually a secondary protocol, the requirements need to be analyzed. However, they may need to be considered for transport only.
  - **Communication lines**: Communication lines support the previously described protocols. Special interfaces may be required to connect to the AS/400e system, remote controllers, personal computers, or routers. The type and speed of the line depends on the requirement of performance and response times. Considerations for such facilities include:
    - *Leased*: Private set of wires from the telephone company. These wires can be:
      - **Point-to-point**: Analog or digital
      - **Multi-point**: Phone company provides the connection of multiple remote sites through the central telephone offices
    - *Integrated Services Digital Network (ISDN)*: Basic or primary access
    - *Frame relay*: An internal standard. The re-routing of traffic is the responsibility of the frame relay provider. For a fail-safe network, all endpoints should have a dial backup or ISDN connection.
    - *Virtual Private Networks (VPN)*: Utilize a service provider that provides encryption and uses the Internet network for the transmission of data to the other site. VPN allows secure transfers over TCP/IP (IPSec) and multiprotocol (L2TP) networks.
    - *Multiprotocol Switched Services*: ATM or LAN
- **Hardware**: A factor that may limit the speed in which communication lines are connected. These factors include:

  - **Controllers**: Manage connections for legacy terminals and personal computers with twinax support cards
  - **Bridge**
  - **Routers**: Allows encapsulation of SNA data into TCP/IP and routes it over a communication network
  - **Frame Relay Access Devices (FRADs)**: Used to buffer SNA devices from a frame relay network. It also channels SNA, BSC, Asynch, and multiprotocol LAN traffic onto a single frame relay permanent virtual circuit and eliminates the need to have separate WAN links for traditional and LAN traffic.
  - **Terminal Adapter**: Used in an ISDN network to buffer the device from the physical connection
  - **Modems**: Used for analog lines
  - **DSU and CSU**: Used for digital lines
  - **Switches**: WAN switches for networks carrying high volumes of traffic at a high speed. These are networks that cope with both legacy and new emerging applications in a single network structure. The network mixes data, voice, image, and video information, and transports different traffic types over a single bandwidth channel to each point.
- **Software**: When the network consists of LANS, routers, and communication lines, a management tool such as Operation Control/400 with Managed System Services/400, Nways Workgroup Manager, or IBM Netfinity is advised to help manage the network.

From routers to remote controllers, review all network components. If there are critical requirements for remote access, you must provide alternative network paths. This can be as simple as a dialup link or as complex as a multi-path private network.

6.4 Testing and single point of failure

The problem with managing a complex computing system is that anything can go wrong with any components at any time. In any development shop, applications are tested. Network, hardware, and external link recovery should be tested with the same emphasis. Just as planning for redundancy, recovery, are a must for high availability, so is testing possible scenarios.

In a highly available environment, testing is particularly more critical. By employing a highly available solution, a business makes a serious commitment to exceptional levels of reliability. These levels not only rely on hardware and software, but they can only be achieved with stringent problem and change management. Identified problems must be replicated and a change must be put in place and tested before further presenting a further production risk.

Testing should involve obvious things (such as hardware, network, and applications), and less obvious components (such as the business processes associated with the computer systems, facilities, data, and applications from external providers, as well as accurately documented job responsibilities). Consider your implementation test environment and your ongoing problem or change test environment when implementing a highly available solution. A

second system or separate LPAR partition can be used for operating system and application test environments.

Figure 16 illustrates a very simple HA environment. There is a two-node cluster or replication environment. A separate system is available for development that has no links to the cluster. Development cannot test changes in a clustered environment. To enable tested changes to be made to the cluster, a fourth system is added. Changes can then be tested on the development systems before moving these changes into production.

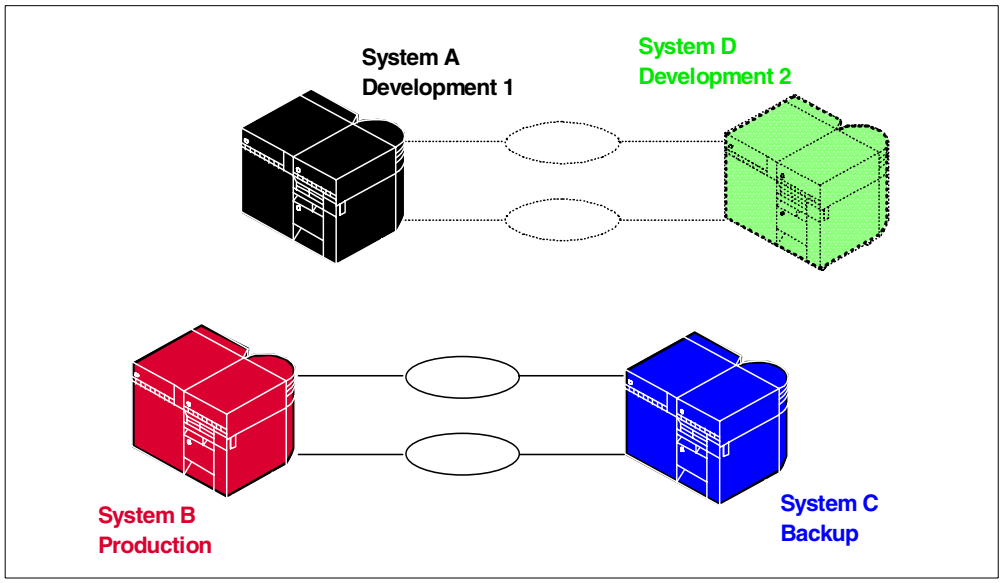


Figure 16. Cluster test scenario

Note: Figure 16 illustrates a basic customer setup. The routers, LANs, WANs, etc., required to simulate the real environment are not shown.

Creating a separate cluster with two smaller systems meets most testing needs. However, some hardware or peripherals may not work with smaller systems. The production environment can only truly be tested with actual and careful function duplication.

Testing needs to involve a planned switchover, a failover (unplanned switchover), the rejoin of the systems, and adding a system to the cluster. Be sure to re-test these scenarios after each system upgrade and any major change to any of the cluster components.

A single network adapter card in a server system that works in a client/server environment is a single-point-of-failure (SPOF) for this server. Likewise, a single SCSI adapter connecting to an external storage system is a SPOF. If a complete server fails within a group of several servers, and the failed server cannot be easily and quickly replaced by another server, this server is an SPOF for the server group or cluster.

The straightforward solution is that adapter cards can be made redundant by simply doubling them within a server and making sure a backup adapter becomes active if the primary one fails.

CPU, power supplies, and other parts can be made redundant within a server too. This requires that the backup adapter becomes active if the primary one fails. These components can also be made redundant by requiring special parts that are not very common in the PC environment and thus quite expensive. However, in cooperation with a software agent (the HA software), two or more servers in an HA cluster can be set up to replace each other in case of a node outage.

When cluster nodes are placed side-by-side, a power outage or a fire can affect both. Consequently, an entire building, a site, or even a town (earthquakes or other natural disasters) can be an SPOF. Such major disasters can be handled by simply locating the backup nodes at a certain distance from the main site. This can be very costly and IT users must be careful to evaluate their situation and decide which SPOFs to cover.

General considerations for testing a high availability environment are described in Appendix F, “Cost components of a business case” on page 169.

6.5 Hardware switchover

A hardware switchover may occur in case of a planned role swap or an unplanned role swap (for example, a system outage). The role swap typically involves 5250 device switching from the failed system to the backup system.

Note: A hardware switchover can also be called a role swap. Figure 17 shows an example IP address assignment for two systems in a cluster preparing for a hardware switchover.

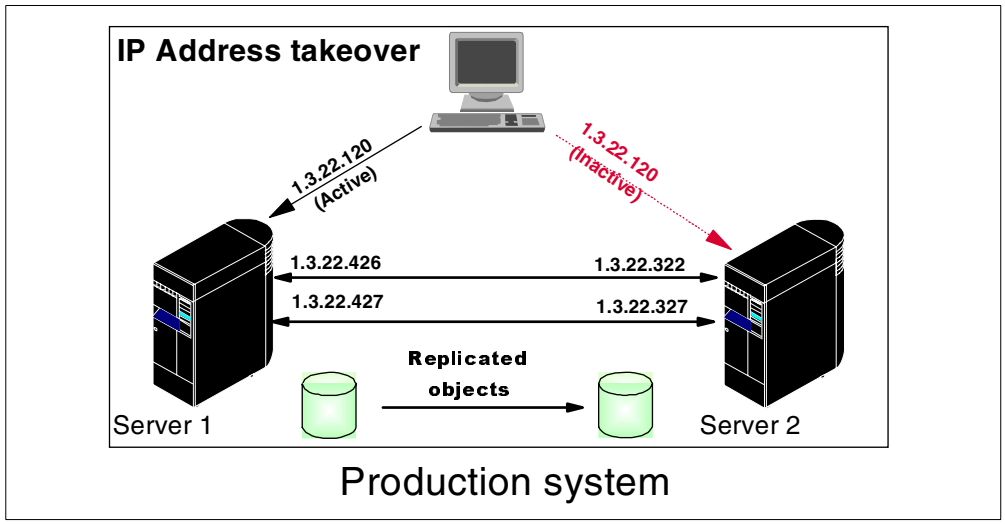


Figure 17. IP address takeover between two systems in a cluster

In a clustered environment, the typical tasks for enabling a hardware switchover are:

1. Quiesce the system:
   a. Remove users from the production system.
   b. Ensure all jobs in the production job queues have completed.
   c. Hold those job queues.

d. Check synchronization of database transactions.
e. End subsystems.

2. End high availability applications on the source machine.

   Make sure all journal receiver entries are complete.

3. End high availability applications on the target machine.
4. Switch between the source and target systems.
5. Switch the network to the target system.
6. Start application and transaction mirroring in reverse mode on the target system.
7. Connect users to the target system.
8. Make necessary changes and updates to the source system and start that system.
9. Switch the roles of the source and target systems again.
10. Switch the network.
11. Start mirroring in normal model.

Note: Switchover capabilities are enhanced at V5R1. However, they are not covered in this Redpaper.

6.6 Network capacity and performance

As business requirements have evolved, a greater dependency has been put on information technology (IT) strategies to remain competitive in the marketplace. More and more, the *marketplace* means an e-business environment. Network performance directly corresponds to business performance. This increased reliance on the network creates the need for high availability within the infrastructure.

New and evolving applications, such as interactive white boarding, video conferences, and collaborative engineering, have significantly increased the need for bandwidth capacity. In addition to bandwidth requirements, these applications create very different traffic patterns from traditional client/server applications. Many of these new applications combine voice, video, and data traffic and have driven the convergence of these infrastructures. As this convergence occurs, the volume of time-sensitive traffic increases. Instantaneous rerouting of traffic must occur to maintain the integrity of the application.

Reliability and fault tolerance are critical to maintain continuous network operations.

6.7 HSA management considerations with networking

Networking management is associated with network performance, particularly in regards to HSA and Continuous Operations support and in a mission-critical application environment. After the network is set up, monitor it to be sure it runs at maximum efficiency.

The primary concerns of continuity are associated with a server's communications pointers to other servers and their respective clients. When networks are consolidated, a common complication is the duplication of network addresses for existing devices. Each of these potential problem areas are readily identified with proper network management tools.

6.7.1 Network support and considerations with a HAV application

Many AS/400 application environments take advantage of several OS/400 communication facilities. The primary concern with these facilities is not so much involved with day-to-day operation activities but is more likely to be involved with the role swap (redirection) of the database function from the production system to a backup system. Moving executing jobs from one physical AS/400 system unit to another requires that pointers managed for these systems are adjusted in a quick and efficient manner. Several communications facilities are involved:

- OptiConnect
- TCP/IP
- SNA

Each facility utilizes certain name attributes or address pointer conventions to indicate another AS/400 system's presence in a network. Implementing continuous operations support requires that the name attributes or pointer conventions be changed to reflect that the current state of business is operating on another AS/400 system.

Scenario with Tivoli network management in action
This section illustrates a fictitious example of what a network management tool can do.

It is noon, the peak traffic time for an international travel agency Web site. Thousands of customers worldwide access the site to book airline, hotel, and car reservations. Without warning, the Web site crashes, literally shutting down the agency's lucrative e-business trade and closing the doors to customers. With each passing minute, tension mounts because the company stands to lose millions of dollars in revenue.

Tivoli instantly sifts through enormous amounts of data for the root cause of the problem. Moments later, it pinpoints the source of the problem: a failing router.

Immediately, the company's e-business traffic is automatically rerouted through an auxiliary router. Meanwhile, system administrators recycle the router and resolve the problem. Downtime is minimized, customers remain satisfied, and a near-disaster is averted. The agency is back online and back in business.

This example illustrates how a network management tool like Tivoli improves system availability.

6.8 Bus level interconnection

Bus level interconnetion consists of hardware and software support between two AS/400 systems at the application level. This path provides a high performance bandwidth for client systems requiring local database access time. By using this feature of the AS/400 system, many customers have been able to use multiple systems to work as a single application image to clients. The SAP R/3

application, with its three-tier configuration and OptiConnect, is an example of how bus level interconnection is used for this purpose.

Figure 18 depicts a three-tiered configuration showing the bus level interconnection arrangement with two bus owners (referred to as a *hub*). This arrangement provides redundancy for the shared bus path.

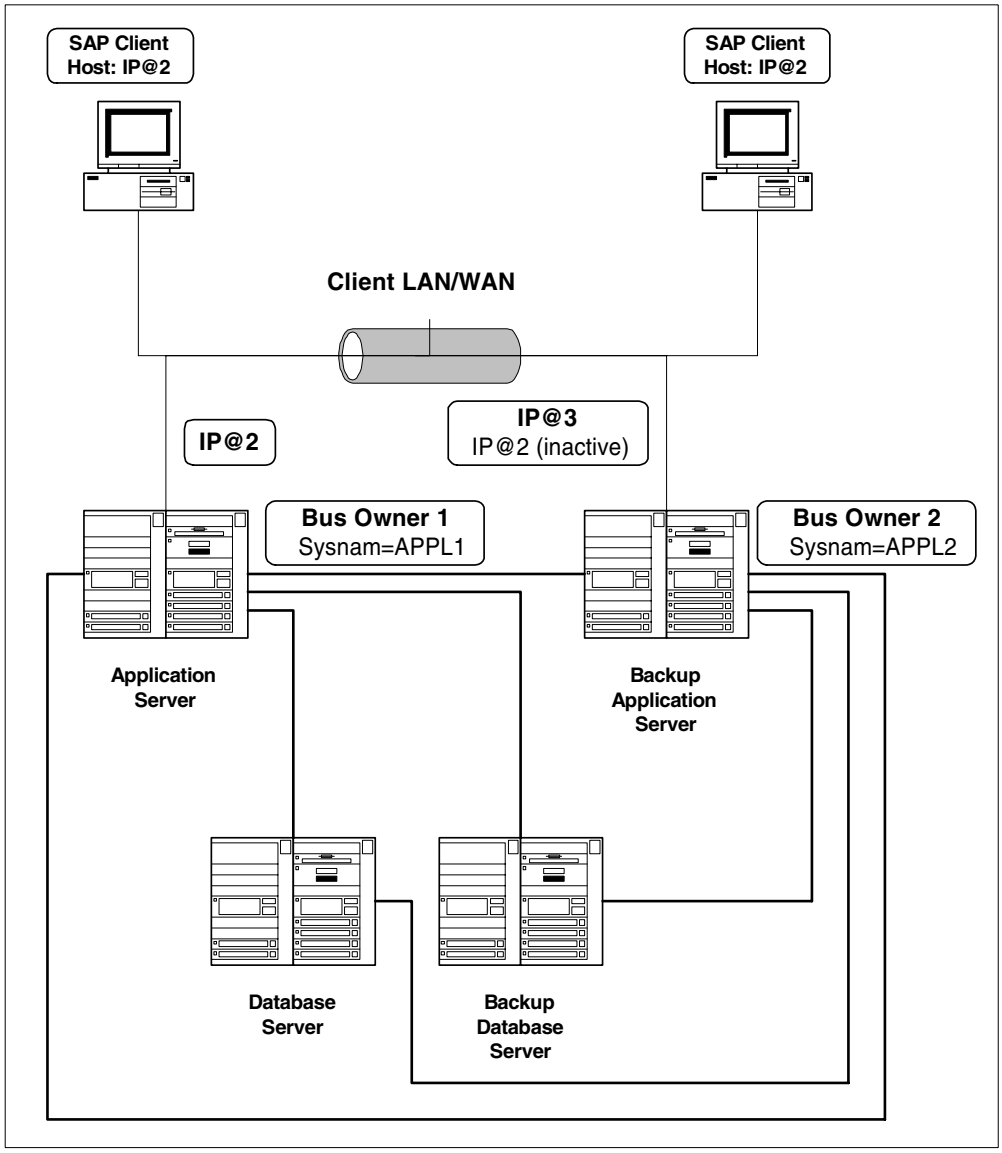


Figure 18. Three-tiered configuration: OptiConnect assignments

In a shared bus redundancy arrangement, it is not necessary for the application servers to be the bus owners because there is no technical reason why the database servers can't assume that role. The only rule you must adhere to with this arrangement is that either application or database servers must perform this role as a pair. You can't have one application server and one database server assume the bus owner role in the redundancy arrangement. If you choose a primary application and a database server for this assignment, there may come a time when both those systems must go through some sort of dedicated process or a power sequence.

This operation can not be allowed to occur because any DDM and SQL procedures through OptiConnect (for example) cease and would, therefore, affect availability of the backup systems when continuous operations is required for the application user. When the bus owner goes to a restricted state or power sequence, all OptiConnect traffic between remote systems stops.

Note: HSL OptiConnect is introduced at V5R1. However, it is not covered in this Redpaper.

6.8.1 Bus level interconnection and a high availability solution

Besides providing high-speed data throughput, bus level interconnection allows OMS/400 to transport data to the backup system efficiently and quickly so that exposure to data loss is very minimal. One shared bus between an application server and two database server systems can accommodate the data requests from the application server. This is done on behalf of the clients to the primary database server while concurrently supporting the data mirroring path between the primary database server and backup database server. However, better resiliency in this example would be provided if both database servers each supported a shared bus to the application server providing a dual bus path (also referred to as shared bus redundancy).

OMS/400 supports the use of OptiConnect for data replication to the backup system. When configuring your application database library in OMS/400, the specification for Opticonnect requires only the System Name from the OS/400 Network Attribute System Name. All pointers to the remote system and the apply processes are built by the configuration process.

6.8.2 TCP/IP

This protocol is also supported by OMS/400. When OptiConnect cannot be used, or is not in operation, this path can also serve as a route for data replication. Primarily, this path is used by SAM/400 for verification of the primary database server's operation and signals the redirection of the primary database system's IP address to the backup database server system in the event of an unplanned outage. In the application environment, it is the path that the client uses for application requests from the application server and the path used between the application server and database server to access stream files prevalent in this environment (also called the *access network*).

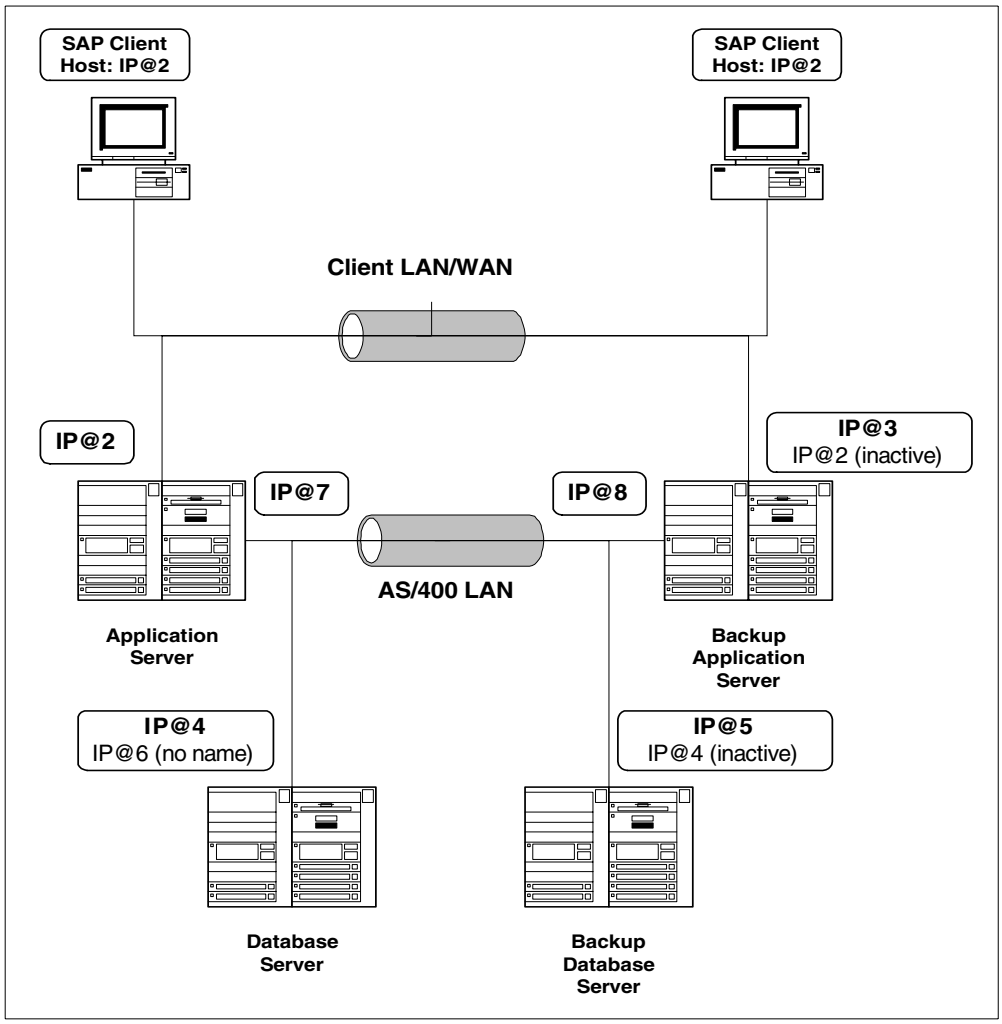


Figure 19. Three-tiered configuration: IP assignments

Figure 19 shows the suggested ethernet LANs for supporting client traffic through the access network (Client LAN/WAN) path. This occurs while operations, development, and SAM/400 use the AS/400 LAN for additional traffic for testing the availability of the production system.

Note: In relationship to Figure 19, the OptiConnect arrangement has been removed from the diagram to keep it uncluttered. However, you can still assume that all four AS/400 systems are connected in that manner.

The LANs can be bridged for network redundancy. However, security would have to be implemented to keep client traffic out of the AS/400 LAN because of direct accessibility to the database servers. It should be noted that SAM/400 can use all of these protocols in combination to test the validity of an unplanned outage.

Figure 19 shows several IP address assignments. This is designed to keep the client interface consistent with one address/host name. In the event of a role swap with an application server, the client operator does *not* need to be concerned with configuration tasks and does not need to use a second icon on their desktop for access to a backup system.

For the application server that must use a backup database server, it also uses the same IP address as that of the primary database server. The role swap procedures outlined in SAM/400 manages the ending and starting of different interfaces. This option facilitates the identification of each system in its new role throughout the LAN. Note that the respective backup systems have inactive IP addresses (as reflected in the interface panel of the TCP configuration menu). These addresses are started during the role swap procedure when the backup system becomes the production environment.

The additional IP address for the primary database server is not generally assigned a host name which is the case with other addresses. This ensures that the address is not being used by the clients through the access network. The main purpose is for the reversed role the primary database server plays when it has stopped operating (for planned or unplanned outages) and it must now be synchronized with the backup database server. Naturally, for continuous operations, the business moves to the backup database server and the database located there begins processing all transactions while the primary database server system is being attended to. After awhile, the primary database server files become aged and must be refreshed with the current state of business before returning to operation.

One method is to save the files from the backup system to the production system. However, this method would impact the user's availability while they are still using the backup database server. OMS/400 is designed to reverse the roles of the primary and backup system to reversed target and reversed source respectively. That means the backup system can capture and send the database changes back to the production system so that it can catch up and be current with the business operation.

The function that the additional IP address plays here is that its allows the system to connect to the AS/400 LAN (as shown in Figure 19) without having to use its normal interface (at this time, it is inactive) when TCP services are required. All this occurs while the business is still using the backup system to run its applications. Once the systems are equal, the user can then return them to their original roles by scheduling a planned role swap. Usually, you would select a time of day when activity is low or when work shifts are changing.

Chapter 7. OS/400: Built-in availability functions

This chapter reviews existing OS/400 functions, specifically those that enhance system availability, databases, and applications. These functions form the foundation of the AS/400 system.

Note: This chapter only summarizes these functions. *The System Administrator's Companion to AS/400 Availability and Recovery,* SG24-2161, provides more details on OS/400 functions designed for availability.

This chapter also outlines clustering and LPAR. Introduced in OS/400 V4R4, these features provide system redundancy and expand availability and replication options.

7.1 Basic OS/400 functions

Some availability functions for the AS/400 system were architected into the initial design (some were carried over from the preceding System/38). Some of these functions include:

- Auxiliary storage pools (ASPs)
- Journaling
- Commitment control

These functions were delivered when the system main storage and processing power were small in comparison to the resource needs of the applications that enabled them. Therefore, many applications were developed without these functions.

As these applications grew in functionality and user base, the task of enabling the functions grew quickly. Now, newer functions are available to provide nearly 100% availability. These new functions are founded on the historic functions and are not enabled by many applications. Therefore, application developers must revisit their applications and enable these historic functions to enable the new functions.

ASPs are discussed in Chapter 5, "Auxiliary storage pools (ASPs)" on page 63. Journaling and commitment control are discussed in this section.

7.1.1 Journaling

Journals define which files and access paths to protect with journal management. This is referred to as journaling a file or an access path. A journal receiver contains journal entries that the system adds when events occur that are journaled, such as changes to database files. This process is shown in Figure 20 on page 88.

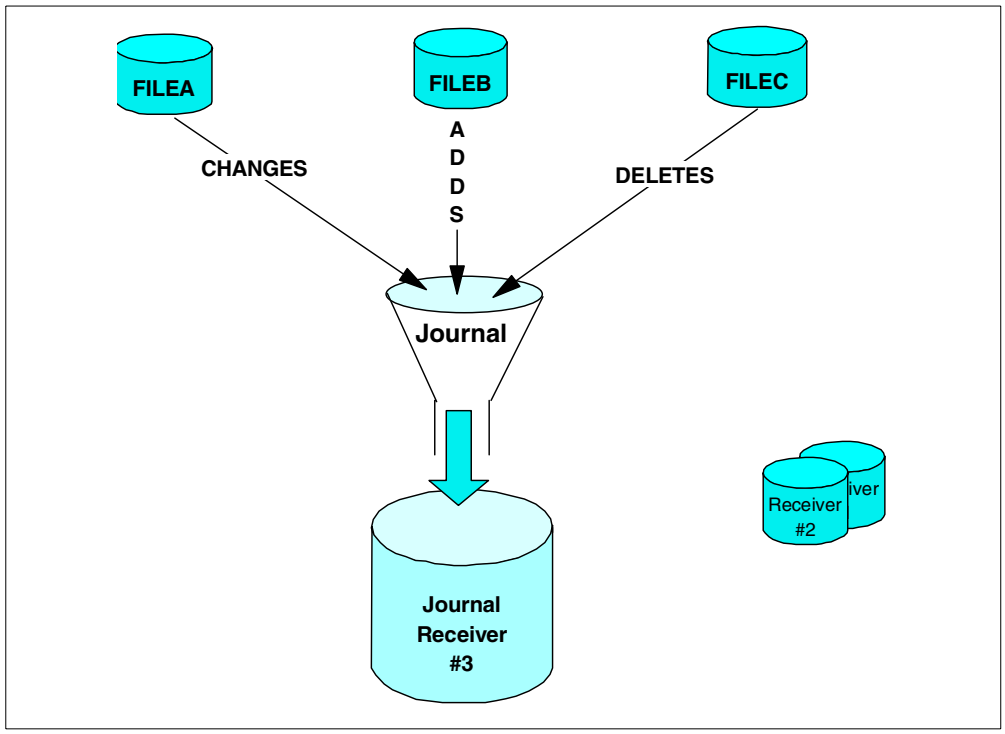


Figure 20. The journaling process

Journaling is designed to prevent transactions from being lost if your system ends abnormally or has to be recovered. Journal management can also assist in recovery after user errors. Journaling provides the ability to roll back from an error to a stage prior to when the error occurred if both the before and after images are journaled.

To recover, the restore must be done in the proper order:

1. Restore your backup from tape.
2. Apply journaled changes.

We recommend that you keep a record of which files are journaled.

Use journal management to recover changes to database files that have occurred since your last complete save.

A thorough discussion of journaling is beyond the scope of this chapter. For more information about journaling, refer to *OS/400 Backup and Recovery*, SC41-5304.

7.1.2 Journal receivers with a high availability business partner solution

Business partner high availability solutions incorporate the use of journaling. A description of journal receivers with OMS/400 and ODS/400 is explained in this section.

Journal receivers are used by OMS/400 and ODS/400. OMS/400 transmits the image of database records recorded there for use by the apply jobs on the target database server. ODS/400, on the other hand, uses the entries recorded in the OS/400 audit journal (QAUDJRN) to tell it what object operations must be performed on the target AS/400 system. The issue with journal receivers is that,

when transactions occur in the applications, and objects are manipulated, the resulting entries placed in the receivers make them grow in size until they reach the maximum threshold of 1.9 GB.

Reaching the maximum threshold must be avoided because the business applications and system functions that use the journaled objects cease operation. They will not resume operation until the filled receiver has been detached from the journal and a new receiver is created and attached.

Typically, the AS/400 user can take advantage of the system's ability to manage the growth and change of receivers while the applications are running. Or, they may elect to write their own management software. However, either solution overlooks one important aspect: synchronization with the reader function that transmits the journal entry to a backup system (OMS/400) or performs an object function (ODS/400) on behalf of the backup system. If, for any reason, the production system cannot communicate the data and object changes to the backup system, no receiver can be deleted until replication resumes and all journal entries have been processed by the high availability application.

Therefore, if the AS/400 user elects one of the first two options to manage receivers, they could inadvertently delete those receivers before their contents were interrogated by high availability software for replication and synchronization purposes.

Note

In such a case, Vision Suite recommends the use of its Journal Manager. This Vision Suite feature changes journal receivers based on a policy established by the user. It also coordinates between the reader jobs for replication and the user's requirements to free auxiliary storage space and save receivers offline. Once it has been implemented, the user's interface for this requirement simply observes the Work with Disk Status (WRKDSKSTS) and monitors the user auxiliary storage pool (ASP) utilization threshold.

Placing journal receivers in user ASPs minimizes the impact journaling may place on some application environments, especially if it has never been used for a given application environment. Since the write I/O changes from asynchronous (with regards to the program execution) to synchronous (where the program producing the write activity must actually wait for the record to be written to the journal receiver) latency is introduced and program execution may increase elapsed time. This result can be seen in batch applications that produce many significant write operations to a database being journaled.

Using user ASPs only for the journal receivers allows for quick responses to the program from the DASD subsystems. The only objects being used in that ASP are the journal receivers and the most typical operation is write I/O. Therefore, the arms usually are positioned directly over the cylinders for the next contiguous space allocation when a journal receiver extent is written to the disk.

Obviously, following this recommendation places more responsibility on the user for managing receivers and the DASD space they utilize. The associated journal must remain in the same ASP as the database it is recording. To implement the user ASP solution, create a new library in the user ASP and, during the next

Change Journal (CHGJRN) operation, specify the new receiver to be qualified to that library.

7.2 Commitment control

Commitment control is an extension of journal management. It allows you to define and process a group of changes to resources, such as database files or tables, as a logical unit of work (LUW). Logically, to the user, the commitment control group appears as a single change. To the programmer, the group appears as a single transaction.

Since a single transaction may update more than one database file, when the system is in a network, a single transaction may update files on more than one system. Commitment control helps ensure that all changes within the transaction are completed for all affected files. If processing is interrupted before the transaction is completed, all changes within the transaction are removed.

Without commitment control, recovering data for a complex program requires detailed application and program knowledge. Interrupted programs cannot easily be re-started. To restore the data up to the last completed transaction, typically a user program or utility, such as a Data File Utility (DFU), is required to reverse incomplete database updates. This is a manual effort, it can be tedious, and it is prone to user error.

Commitment control ensures that either the entire group of individual changes occur on all participating systems, or that none of the changes occur. It can assist you with keeping your database files synchronized.

7.2.1 Save-while-active with commitment control

Using the save-while-active function while commitment control processing is active requires additional consideration. When an object is updated under commitment control during the checkpoint processing phase of a save-while-active request, the system ensures that the object is saved to the media at a commitment boundary. All objects that have reached a checkpoint together are saved to the media at the same common commitment boundary.

It is important to make sure that all performance considerations have been correctly implemented in this situation. Otherwise, the system may never be able to reach a commitment boundary. It may not be able to obtain a checkpoint image of the objects to be saved. Procedures need to be specified to ensure that all of the objects reach a checkpoint together and all of the objects are saved in a consistent state in relationship to each other. If the checkpoint versions of the objects are not at an application boundary, user-written recovery procedures may still be necessary to bring the objects to an application boundary.

Refer to *OS/400 Backup and Recovery*, SC41-5304 for details on coding for commitment control and save-while-active.

7.3 System Managed Access Path Protection (SMAPP)

An access path describes the order in which records in a database files are processed. A file can have multiple access paths if different programs need to see the records in different sequences. If your system ends abnormally when access

paths are in use, the system may have to rebuild the access paths before you can use the files again. This is a time-consuming process. To perform an IPL on a large and busy AS/400 system that has ended can take many hours.

You can use journal management to record changes to access paths. This greatly reduces the amount of time it takes the system to perform an IPL after it ends abnormally.

Access path protection provides the following benefits:

- Avoids rebuilding access paths after most abnormal system ends
- Manages the required environment and makes adjustments as the system changes if SMAPP is active
- Successful even if main storage cannot be copied to storage Unit 1 of the system ASP during an abnormal system end
- Generally faster and more dependable than forcing access paths to auxiliary storage for the files (with the FRCACCPTH parameter)

The disadvantages of access path protection include:

- Increases auxiliary storage requirements
- May have an impact on performance because of an increase in the activity of the disks and processing unit
- Requires file and application knowledge for recovery. There is a small additional processor overhead if *RMVINTENT is specified for the RCVSIZOPT parameter for user-created journals. However, the increase in storage requirements for access path journaling is reduced by using *RMVINTENT.
- Normally requires a significant increase in the storage requirements for journaling files. The increase with SMAPP is less than when access paths are explicitly journaled.

Two methods of access-path protection are available:

- System management access-path protection (SMAPP)
- Explicit journaling of access paths

An access path (view) describes the order in which records in a database file are processed. A file can have multiple access paths if different programs need to see the records in different sequences. If your system ends abnormally when access paths are in use, the system may have to rebuild the access paths before you can use the files again. This can be a time-consuming process, since an IPL on a large busy server that had ended abnormally may take many hours. Two methods of access-path protection are available:

- OS/400 System Managed Access Path Protection (SMAPP)
- Explicit journaling of access paths

7.4 Journal management

You can use journal management to recover the changes to database files or other objects that have occurred since your last complete save. Use a journal to define what files and access paths you want to protect with journal management. This is often referred to as journaling a file or an access path. A journal receiver

contains the entries (called journal entries) that the system adds when events occur that are journaled, such as changes to database files, changes to other journaled objects, or security-relevant events.

Use the remote journal function to set up journals and journal receivers on a remote AS/400 system. These journals and journal receivers are associated with journals and journal receivers on the source system. The remote journal function allows you to replicate journal entries from the source system to the remote system.

The main purpose of journal management is to assist in recovery. You can also use the information that is stored in journal receivers for other purposes, such as:

- An audit trail of activity that occurs for database files or other objects on the system
- Assistance in testing application programs. You can use journal entries to see the changes that were made by a particular program.

Figure 21 shows the steps involved for journaling.

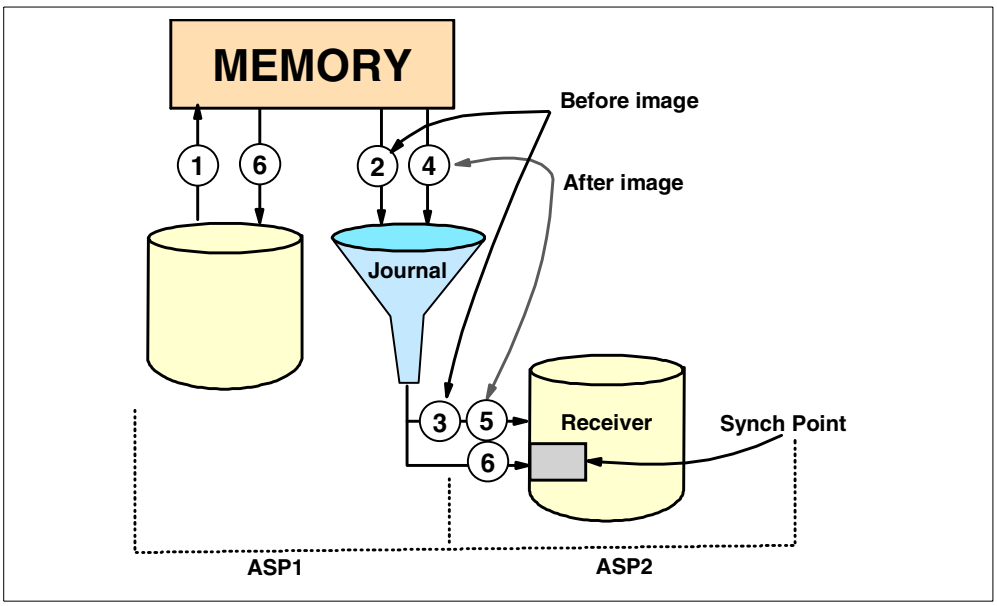


Figure 21. The Journaling Process

7.4.1 Journal management: Benefits

Benefits of journal management can include:

- A reduction in the frequency and amount of data saved
- Improved ability and speed of recovery from a known point to the failure point
- Provides file synchronization if the system ends abnormally

7.4.2 Journal management: Costs and limitations

Disadvantages of journal management include:

- An increase in auxiliary storage requirements
- Can have an impact on performance because of an increase in the activity of disks and the processing unit

- Requires file and application knowledge for recovery

Refer to the *OS/400 Backup and Recovery*, SC41-5304 for further information.

7.5 Logical Partition (LPAR) support

In an n-way symmetric multi-processing iSeries or AS/400e server, logical partitions allow you to run multiple independent OS/400 instances or partitions, each with their own processors, memory, and disks.

Note: With OS/400 V5R1, a single processor can be *sliced* for sharing across partitions.

You can run a cluster environment on a single system image. With logical partitioning, you can address multiple system requirements in a single machine to achieve server consolidation, business unit consolidation, and mixed production and test environments.

Figure 22 shows an LPAR configuration with resources shared across partitions.

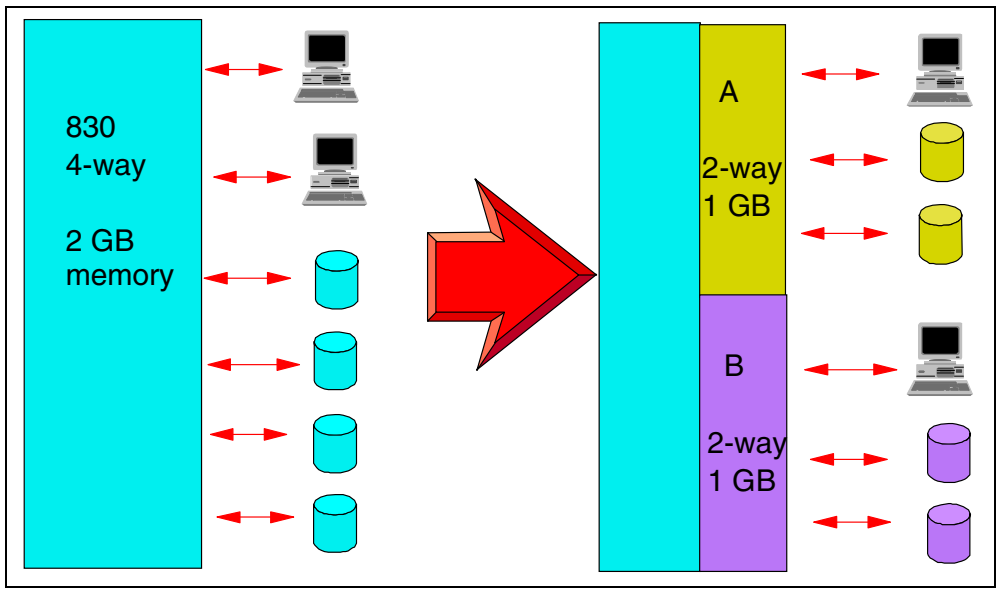


Figure 22. Example LPAR configuration

Each logical partition represents a division of resources in your AS/400e system. Each partition is logical because the division of resources is virtual rather than physical. The primary resources in your system are its processors, memory (main storage), I/O buses, and IOPs.

An LPAR solution does not offer a true failover capability for all partitions. If the primary partition fails, all other partitions also fail. If there are multiple secondary partitions backing each other up, they have the capability to failover between partitions. These secondary partitions are nodes and are a cluster solution. However, they are not a separate server implementation. LPAR cannot provide the same level of availability as two or more node cluster solutions.

See 4.12, “LPAR hardware perspective” on page 58, for discussion of LPAR from a hardware perspective.

7.6 Cluster support and OS/400

The ultimate availability solution consists of clustered systems. OS/400 V4R4 introduced clustering support. This support provides a common architected interface for application developers, iSeries software providers, and high availability business partners to use in-building high availability solutions for the iSeries and AS/400 server. The architecture is built around a framework and is the foundation for building continuous availability solutions for both the iSeries and AS/400 servers.

Clustering provides:

- Tools to create and manage clusters, the ability to detect a failure within a cluster, and switchover and failover mechanisms to move work between cluster nodes for planned or unplanned outages
- A common method for setting up object replication for nodes within a cluster (this includes the data and program objects necessary to run applications that are cluster-enabled)
- Mechanisms to automatically switch applications and users from a primary to a backup node within a cluster for planned or unplanned outages

Clustering involves a set of system APIs, system services, and exit programs as shown in Figure 23. Data replication services and the cluster management interface are provided by IBM HABPs.

How clustering works, and planning for clusters, is described in *AS/400 Clusters, A Guide to Achieving Higher Availability*, SG24-5194.

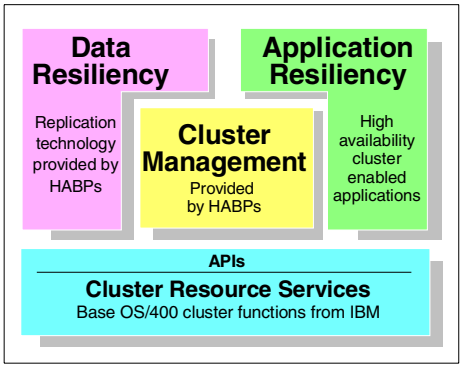


Figure 23. Cluster overview

Chapter 8. Performance

Performance is an availability issue from several view points. First and foremost, to the end user, poor performance can be significantly more frustrating than a system outage. Poor performance can even result in lost sales. An example would be a customer calling to check the availability or price of an item. If the response of the service representative (or a Web-enabled application) is too long, the customer looks elsewhere.

Poor performance can be caused by any number of factors, which can be any computing component between the end-user incoming request and the delivery of the response. The timing is affected by the performance of the communication links, routers, hubs, disk arms (service time), memory, and the CPU. Service agreements can hold both parties accountable, with the parties being the service provider (you the business), and the receiver or requestor of the service (the customer).

Do not delay the planning for performance for your high availability environment until after implementing the system and applications. Plan for it prior to installation. Set your expectations and guidelines accordingly.

It is easy to suggest that, by implementing a backup machine, the spare capacity on the backup can give extra cycles to a sluggish application. This is very rarely the case. Investigating the proposed performance of the new environment should reap dividends during the implementation phase.

This section discusses the implication of performance on the various levels of availability.

Note: Performance ratings of save and restore hardware and software options can be found in the AS/400 Performance Capabilities Reference. This can be accessed online at: `http://www.ibm.com/eserver/iseries/library`

8.1 Foundations for good performance

This section briefly describes the fundamental elements of good performance on the AS/400e servers.

8.1.1 Symmetric multiprocessing (SMP)

In the AS/400e world, SMP has multiple meanings. First and foremost, it is a hardware deployment capability. iSeries processors and some AS/400e processors can be purchased in 2-way, 4-way, 8-way, 12-way, 18-way, and 24-way configurations. In the industry, all the n-way processor configurations are referred to as SMP processor systems.

Due to the architecture of the AS/400 server and OS/400, applications and utilities are able to take advantage of the SMP models without overt programming efforts. However, it is possible to obtain even higher levels of throughput by redesigning batch processes to take advantage of multiple processors.

Another use of the term SMP in the AS/400e world refers to a feature of the operating system called *DB2 Symmetric Multiprocessing for AS/400*. This feature enables a dynamic build of access paths or views for queries (including

OPNQRYF and the query manager) utilizing parallel I/O and parallel processing across all available processors. Plan carefully for the use of this feature because it can significantly increase overall CPU utilization in addition to increased physical I/O operations.

To learn more about the DB2 Symmetric Multiprocessing for AS/400 feature, refer to the *iSeries Handbook,* GA19-5486 and the *Performance Capabilities Reference,* which can be accessed online at:
`http://publib.boulder.ibm.com/pubs/pdfs/as400/V4R5PDF/AS4PPCP3.PDF`

8.1.2 Interactive jobs

Just as you would not implement a major application change without analysis of the potential performance impact, you need to determine the potential impact of implementing high availability on your interactive workload. Ask the following questions:

- What effect will the proposed implementation have on interactive performance?
- If journaling is to be activated, what will its impact be?
- Have you decided to rewrite your applications to ensure data integrity by utilizing commitment control? What will this performance impact be?

The answers to these questions must be analyzed and fed back to the business plan and incorporated into your service level agreements.

8.1.3 Batch jobs

Batch jobs are another key area for high availability. This is one place that backup machines may have a positive effect on the performance of the primary system. You may be able to redirect work from your primary system to the backup system. This is most feasible for read only work. Other types of batch jobs could be very difficult to alter to take advantage of a second system and a major re-write may be necessary.

If you choose to utilize your backup system for read only batch work, make sure that you understand the impact of these jobs on the high availability business partner apply processes. If the work you run on the backup system interferes in any way with these apply processes, you may reduce your ability to switchover or failover in a timely manner.

You need to consider the impact of activating journaling on your batch run times and explore the possibility of incorporating commitment control to improve run time in a journaled environment. This is discussed in more detail in 8.2.3, “Application considerations and techniques of journaling” on page 99.

8.1.4 Database

Consider the types of database networks utilized by your application when understanding performance. Do you have multiple database networks, a single database network, multiple database networks across multiple servers, or a single database network across multiple servers?

As stated, the major performance impact for a database is the start of journaling. Each database operation (except read operations) involves journal management. This adds a physical I/O and code path to each operation.

8.2 Journaling: Adaptive bundling

Journaling's guarantee of recover ability is implemented with extra I/O and CPU cycles. Since V4R2 of OS/400, the technique of adaptive bundling has been used to reduce the impact of these extra I/O operations.

This means that journal writes are often grouped together for multiple jobs, in addition to commit cycles. Refer to Figure 24 for a simple illustration.

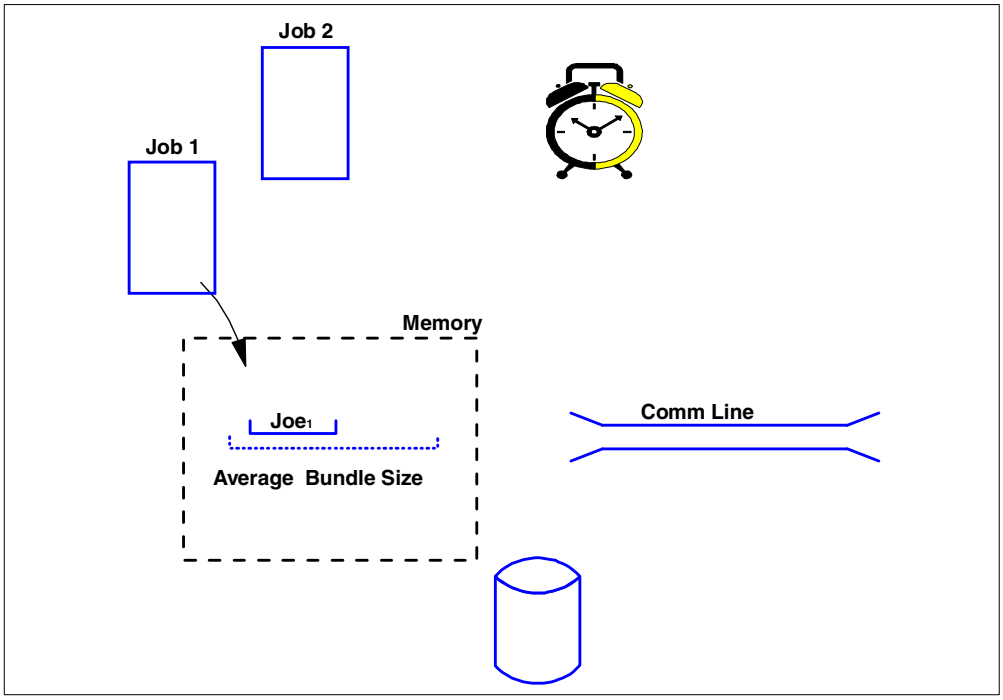


Figure 24. Adaptive bundling

Unless you have taken specific actions, a single batch job can minimally take advantage of adaptive bundling. A high penalty is paid in extra I/O with every record update performed.

It is normal to see an increase in both I/O and CPU utilization after turning on journaling. Even on a well-tuned system, the CPU utilization increase can be as high as 30%. This increase in utilization increases response time. An even higher degradation is common for single threaded batch jobs that do not run commitment control.

Journaling increases the number of asynchronous writes. The effect of these asynchronous writes is shown on the Transition Report of Performance Tools. Modules QDBPUT, QDBGETKY, and QDBGETSQ show evidence of this asynchronous I/O request.

To reduce the impact of journaling:

- Group inserts under a commit cycle
- Group inserts
- Split a batch job into several jobs and run them in parallel

8.2.1 Setting up the optimal hardware environment for journaling

Building a sound hardware environment for your journaled environment can minimize the impact of journaling. Start by creating a user auxiliary storage pool (ASP) that utilizes mirrored protection.

Depending on the amount of storage required for your journal receivers, and the release of OS/400 you are running, you can allocate between 6 and 200 total arms to this user ASP. These disk arms should have dedicated IOPs with at least .5 MB of write cache per arm. Regardless, these arms should be the fastest available on your system.

Note: The total stated number of arms are not “operational” numbers after turning on mirroring.

If you are running a release of OS/400 prior to V4R5, the maximum number of disk arms the system can efficiently use for parallel I/O operations for journaling is 30. In V4R5, if you utilize the *MAXOPT parameter on the receiver size (RCVSIZOPT) keyword, the number of used disk arms increases to 200.

8.2.2 Setting up your journals and journal receivers

Refer to Section 6.2 “Planning and Setting Up Journaling” in the *Backup and Recovery Guide*, SC41-5304, for detailed information. Also keep the following points in mind:

- The journal and journal receiver objects should not be in the same library as the files they are to journal.
- The journal object (*JRN) should not reside in the user ASP of the journal receiver (*JRNRCV) objects.
- Isolate journal receiver writes from system managed access path protection (SMAPP) writes by specifying RCVSIZOPT(*RMVINTENT) on the CRTJRN and CHGJRN commands. In addition to isolating the SMAPP I/O operations to arms dedicated for that activity, your journal receivers will not fill as quickly. The system uses two-thirds of allocated ASP arms for JRNRCV objects and the remaining one-third for SMAPP entries.
- Suppress open and close journal entries by utilizing OMTJRNE(*OPNCLO) on the STRJRNPF command.
- Use system managed receivers by specifying MNGRCV(*SYSTEM) on the CRTJRN and CHGJRN commands to enable better system performance during the change of journal receivers. You can ensure that your business partner package maintains control over the actual changing of journal receivers by specifying a threshold on your journal receivers that is larger than the size specified in the partner package.

  The MNGRCV(*SYSTEM) requires the parameter THRESHOLD be specified on the CRTJRNRCV command.

8.2.2.1 Determining the number of journals and receivers

Generally speaking, you always have multiple journal and journal receivers. Some strategies for determining the number of journals and journal receivers you have include:

- **By application**: To simplify recovery, files that are used together in the same application should be assigned to the same journal. In particular, all the physical files underlying a logical file should be assigned to the same journal. Starting in V3R1, all files opened under the same commitment definition within a job do not need to be journaled to the same journal. If your major applications have completely separate files and backup schedules, separate journals for the applications may simplify operating procedures and recovery.
- **By security**: If the security of certain files requires that you exclude their backup and recovery procedures from the procedures for other files, assign them to a separate journal, if possible.
- **By function**: If you journal different files for different reasons, such as recovery, auditing, or transferring transactions to another system, you may want to separate these functions into separate journals. Remember, a physical file can be assigned to only one journal.

If you have user ASPs with libraries (known as a library user ASP), all files assigned to a journal must be in the same user ASP as the journal. The journal receiver may be in a different ASP. If you place a journal in a user ASP without libraries (non-library user ASP), files being journaled must be in the system ASP. The journal receiver may be in either the system ASP or a non-library user ASP with the journal. See the section titled “Should Your Journal Receivers Be in a User ASP?” in the *Backup and Recovery Guide*, SC41-5304, for more information about the types of ASPs and restrictions.

Remember to consult the *Backup and Recovery Guide*, SC41-5304 for restore or recovery considerations when setting up your environment. Even though you set up this environment to minimize the need to ever fully restore your system, you may have to partially restore within your own environment or fully restore if you take advantage of the Rochester Customer Benchmark Center or a disaster/recovery center.

8.2.3 Application considerations and techniques of journaling

Database options that have an impact on journaling and system performance are:

- The force-write ratio (FRCRATIO) parameter for physical files that are journaled. This allows the system to manage when to write records for the physical file to disk because, in effect, the journal receiver has a force-write ratio of one.
- Record blocking when a program processes a journaled file sequentially (SEQONLY(*YES)). When you add or insert records to the file, the records are not written to the journal receiver until the block is filled. You can specify record blocking with the Override with Database File (OVRDBF) command or in some high-level language programs. This is a standard and good performance practice that significantly helps the performance of journaling too.

- Use OMTJRNE(*OPNCLO)) to reduce the number of journal entries. If you choose to omit open journal entries and close journal entries, note the following considerations:
  - You cannot use the journal to audit who has accessed the file for read only purposes.
  - You cannot apply or remove journal changes to open boundaries and close boundaries using the TOJOBO and TOJOBC parameters.
  - Another way to reduce the number of journal entries for the file is to use shared open data paths. This is generally a good performance recommendation regardless of journaling activity.
- Utilize the Batch Journal Caching PRPQ. This offering:
  - Forces journal entries to be cached in memory for most efficient disk writes
  - Is designed to reduce journaling's impact on batch jobs
  - Is selectively enabled

Additional information about the Batch Journal Caching PRPQ can be found in Appendix C, “Batch Journal Caching for AS/400 boosts performance” on page 153.

8.3 Estimating the impact of journaling

To understand the impact of journaling on the capacity of a system, consider the processes involved. Additional overhead is involved for disk and CPU activity and additional storage is required in preparation for potential recovery.

8.3.1 Additional disk activity

Consider that each row updated, added, or deleted has either one or two journal entries. While this is an asynchronous I/O operation, your disk arm response time can increase. This causes degradation to your production workload. Under certain circumstances, these asynchronous I/O operations become synchronous I/O operations and cause your application to wait for them to complete before they can continue.

8.3.2 Additional CPU

Each update, add, or delete operation utilizes additional CPU seconds to complete. The ratio of CPU per logical I/O is a key factor in determining the additional CPU required for journaling.

8.3.3 Size of your journal auxiliary storage pool (ASP)

Depending on how accurate you want your estimate of space requirements to be, follow one of the two methodologies explained in this section to estimate your space requirements.

8.3.3.1 Using weighted average record length

Perform the following steps to use a weighted average record length:

1. Determine the average number of entries per time period (number of days or hours) worth of receiver entries you want or need available.

To take advantage of the journal's ability to protect your data and to provide an audit trail, you may want to keep more than a few hours worth of receiver entries for transmission to a secondary system. Once the data is written to the disk, the only expense involved beyond the disk space consumed is the cycles required to retrieve the entries for further analysis.

2. Determine the weighted average record length.

   Add 155 to this weighted average record length.

3. Multiply the results from step 2 by the results from step 1 and divide by 1,024 to determine the KB. Or, divide by 1,048,576 to determine the MB required.

8.3.3.2 Using actual changes logged by file management

The file description contains information useful for calculating storage usage. To utilize this information:

1. Execute a CL program to capture FD information.
2. Execute an RPG program to translate the date to a table for further calculations.
3. Rerun CL in as you did in step 1.
4. Execute an RPG program to add a second set of FD information to a table.
5. Calculate requirements based on information in the file.

8.4 Switchover and failover

Highly available systems involve clustering. When a production system is switched over to the backup system, either for a planned or an unplanned outage, the time required to make this switchover is critical.

To reduce the time involved in this switchover process, consider the networks and performance as explained in the following section.

Networks and performance

The performance of a communications network provides acceptable (or non-acceptable) response time for the end users. Response time provides the perception for the end user of the reliability and availability of the system. In general, to improve performance:

- Avoid multiple layers of communications.
- Avoid communication servers (such as Microsoft SNA server, IBM Communication Server, or NetWare SNA server).
- Use Client Access/400 or IBM eNetwork Personal Communications for AS/400 where possible.
- Use a native protocol instead of ANYNET.

The Best/1 licensed program, a component of 5769-PT1, can capture information on a communication line and predict utilization and response times.

Part 3. AS/400 high availability solutions

Combining the features of OS/400 system and network hardware with AS/400e high availability software produced by IBM business partners is an important method for improving a single systems availability. Part III discusses these components and it also explores the considerations for writing applications to support a highly available environment.

Chapter 9. High availability solutions from IBM

The foundation of all high availability functions is OS/400. The AS/400 development and manufacturing teams continue to improve the AS/400 system for feature, function, and reliability options with each release of OS/400. In particular, the OS/400 remote journal feature enhances high availability solutions by enabling functions below the machine interface (MI) level.

Note: Prior to OS/400 V4R2, remote journal functions were coded into application programs. APIs and commands are available in V4R2 and V4R3 respectively.

Cluster solutions connect multiple AS/400 systems together using various interconnect fabrics, including high-speed optical fiber. Clustered AS/400 systems offer a solution that can deliver up to 99.99% system availability.

For planned or unplanned outages, clustering and system mirroring offer the most effective solution. IBM business partners that provide high systems availability tools complement IBM availability tools with replication, clustering, and system mirroring solutions.

The IBM (application) contribution to AS/400 High Availability Solutions includes the IBM DataPropagator Relational Capture and Apply for AS/400 product. From this point on, this product is referred to as DataPropagator/400. This chapter describes the IBM package and its benefits as a minimal High Availability Solution.

9.1 IBM DataPropogator/400

The IBM solution that fulfills some of the requirements of an HSA solution is DataPropagator/400.

Note: The DataPropagator/400 product was not designed as a high availability solution. In some cases, it can cover the needs for data availability, as discussed in this chapter.

DataPropagator/400 is a state-of-the-art data replication tool. Data replication is necessary when:

- Supplying consistent real time reference information across an enterprise
- Bringing real time information closer to the business units that require access to insulate users from failures elsewhere on the network
- Reducing network traffic or the reliance on a central system
- On-demand access disrupts production or response
- Migrating systems and designing a transition plan to move the data while keeping the systems in sync
- Deploying a data warehouse with an automated movement of data
- Current disaster plan strategies do not adequately account for site-failure recovery

DataPropagator/400 is not a total High Availability Solution because it only replicates databases. It does not replicate all of the objects that must be mirrored for a true High Availability Solution in a dynamic environment.

However, consider DataPropagator/400 for availability functions in a stable environment where the following criteria can be met:

- Only the database changes during normal production on the AS/400 system.
- Such objects as user profiles, authorities, and other non-database objects are saved regularly on the source system and restored on the target system when changed.

In other words, in a stable environment, where only the database changes, replicating the database to a backup system and transferring users manually to this system may be a sufficient availability and recovery plan (Figure 25).

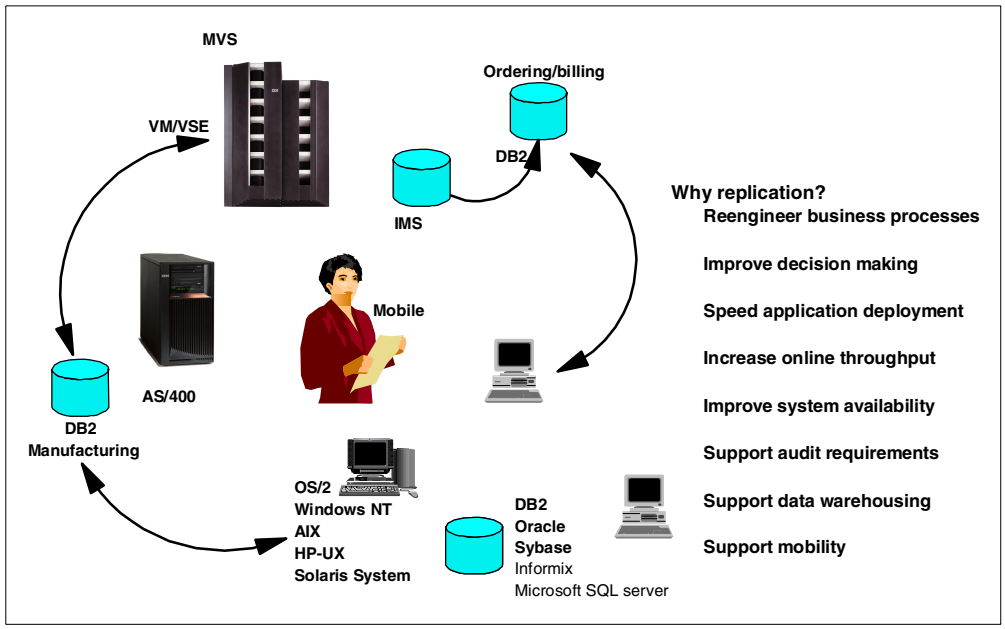


Figure 25. Usage of DataPropagator/400

9.1.1 DataPropagator/400 description

IBM DataPropagator/400 automatically copies data within and between IBM DB2 platforms to make data available on the system when it is needed. The IBM DataJoiner product can be used in addition to the DataPropagator/400 product to provide replication to several non-IBM databases. Immediate access to current and consistent data reduces the time required for analysis and decision making.

DataPropagator/400 allows the user to update copied data, maintain historical change information, and control copy impact on system resources. Copying may involve transferring the entire contents of a user table (a full refresh) or only the changes made since the last copy (an update). The user can also copy a subset of a table by selecting the columns they want to copy.

Making copies of database data (snapshots) solves the problem of remote data access and availability. Copied data requires varying levels of synchronization with production data, depending on how the data is used. Copying data may even be desired within the same database. If excessive contention occurs for data access in the master database, copying the data offloads some of the burden from the master database.

By copying data, users can get information without impacting their production applications. It also removes any dependency on the performance of remote data access and the availability of communication links.

DataPropagator/400 highlights include:

- An automatic copy of databases
- Full support for SQL (enabling summaries, derived data, and subsetted copies)
- During a system or network outage, the product restarts automatically from the point where it stopped. If this is not possible, a complete refresh of the copies can be performed if allowed by the administrators. Also, for example, if one of the components fails, the product can determine that there is a break in sequence of the data being copied. In this case, DataPropagator/400 restarts the copy from scratch.
- Open architecture to enable new applications
- DataPropagator/400 commands that support AS/400 system definitions
- Full use of remote journal support in V4R2

9.1.2 DataPropagator/400 configuration

In the database network, the user needs to assign one or more roles to their systems when configuring the DataPropagator/400 environment. These roles include:

- **Control server**: This system contains all of the information on the registered tables, the snapshot definitions (the kind of data you want to copy and how to copy it), the ownership of the copies, and the captures in reference to registrars and subscribers.
- **Data server**: This contains the source data tables.
- **Copy server**: This is the target system.

Depending on the structure of the company, the platforms involved, and customer preferences, a system in the network can play one or more of these three roles. DataPropagator/400, for example, works powerfully on a single AS/400 system, which, at the same time, serves as the Control, Copy, and Data Server.

9.1.3 Data replication process

With DataPropagator/400, there are two steps to the data replication process:

- **The Capture process**: This is for reading the data.
- **The Apply process**: This is for applying updated data

Figure 26 and Figure 27 on page 108 illustrate these processes.

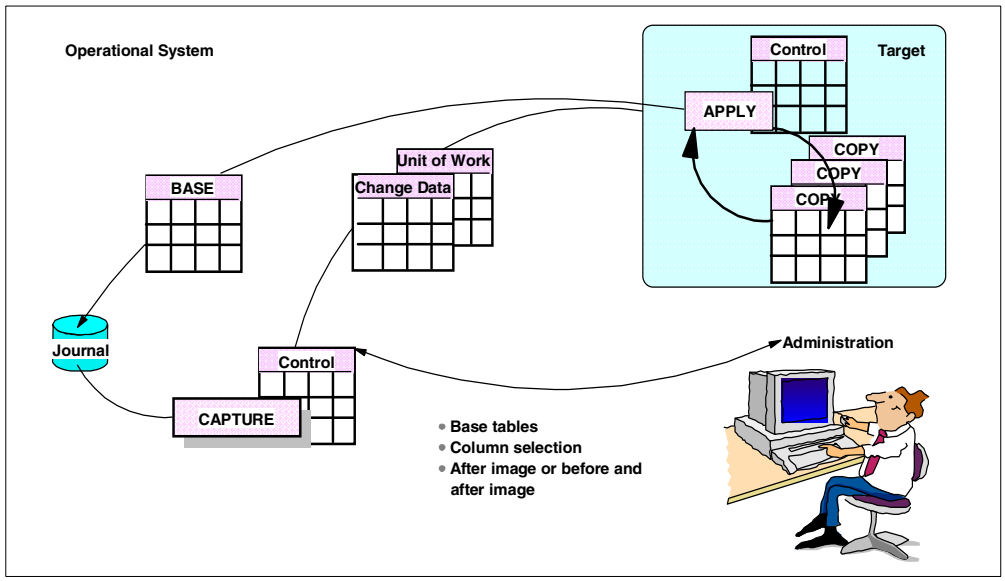


Figure 26. The DataPropagator/400 Capture process

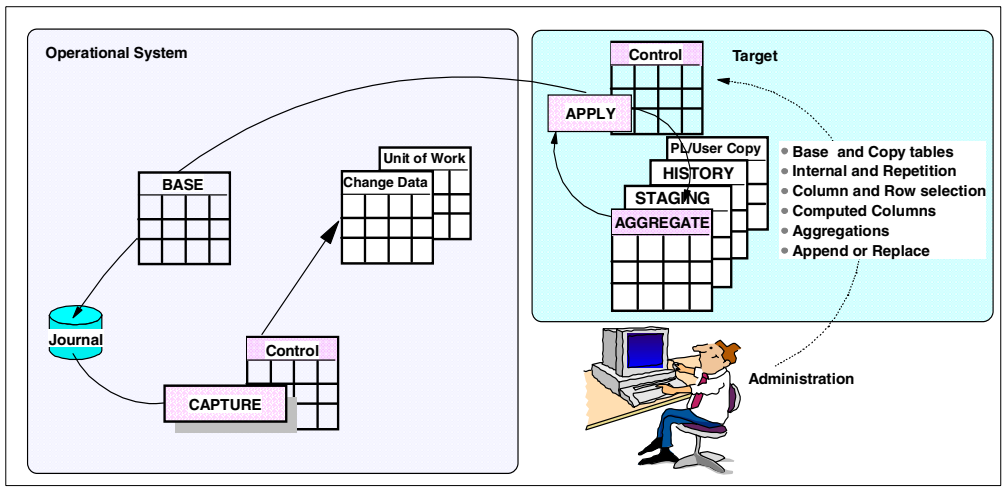


Figure 27. The DataPropagator/400 Apply process

Features included in the Capture and Apply processes include:

- Support for the remote journal function to offload the source CPU
- Automated deletion of journal receivers
- Replication over a native TCP/IP-based network
- Multi-vendor replication with DataJoiner (replication to and from Oracle, Sybase, Informix, and Microsoft SQL Server databases)
- Integration with the Lotus Notes databases

9.1.4 OptiConnect and DataPropagator/400

DataPropagator/400 is based on a distributed relational database architecture (DRDA) and is independent of any communications protocol. Therefore, it uses OptiConnect and any other media without additional configuration.

9.1.5 Remote journals and DataPropagator/400

DataPropagator/400 takes advantage of an operating system's remote journal function. With remote journals, the capture process is run at the remote journal location to offload the capture process overhead from the production system.

The Apply process does not need to connect to the production system for differential refresh because the DataPropagator/400 staging tables reside locally rather than on the production system. In addition, because the DataPropagator/400 product is installed only on the system that is journaled remotely, the production system no longer requires a copy of DataPropagator/400.

9.1.6 DataPropagator/400 implementation

DataPropagator/400 is most beneficial for replicating data to update remote databases. One real-life example of this is a customer in Denmark who had a central AS/400 system and stored all production data, pricing information, and a customer database on it. From this central machine, data was distributed to sales offices in Austria, Germany, Norway, and Holland (each of which operated either small AS/400 systems or OS/2 PCs). Each sales office received a subset of the data that was relevant to their particular office. See 3.1, "A high availability customer: Scenario 1" on page 25, for a description of this customer scenario.

9.1.7 More information about DataPropagator/400

For more information about IBM DataPropagator/400 solutions, refer to *DataPropagator Relational Guide*, SC26-3399, and *DataPropagator Relational Capture and Apply/400*, SC41-5346. Also, visit the IBM internet Web site at:
`http://www.software.ibm.com/data/dbtools/datarepl.html`

Chapter 10. High availability business partner solutions

High availability middleware is the name given to the group of applications that provide replication and management between AS/400e systems and cluster management middleware. IBM business partners that provide high system availability tools continue to complement IBM availability offerings of clustering and system mirroring solutions. Combining clusters of AS/400 systems with software from AS/400 high-availability business partners improves the availability of a single AS/400 system by replicating business data to one or more AS/400 systems. This combination can provide a disaster recovery solution.

This chapter explores the applications provided by IBM High Availability Business Partners (HABPs), DataMirror, Lakeview Technology, and Vision Solutions. This chapter also discusses the options these companies provide to support a highly or continuously available solution.

Note

Some of the software vendors mentioned in this chapter may have products with functions that are not directly related to the high availability issues on the AS/400 system. To learn more about these products, visit these vendors on the World Wide Web. You can locate their URL address at the end of the section that describes their solution.

Note: For customers requiring better than 99.9% system availability, AS/400 clusters with high-availability solutions from an IBM Business Partner are recommended.

10.1 DataMirror Corporation

DataMirror Corporation is an IBM business partner with products that address a number of issues, such as data warehousing, data and workload distribution, and high availability. DataMirror products run on IBM and non-IBM platforms.

The DataMirror High Availability Suite uses high performance replication to ensure reliable and secure delivery of data to backup sites. In the event of planned or unplanned outages, the suite ensures data integrity and continuous business operations. To avoid transmission of redundant data, only changes to the data are sent to the backup system. This allows resources to be more available for production work. After an outage is resolved, systems can be resynchronized while they are active.

The DataMirror High Availability Suite contains three components:

- DataMirror High Availability (HA) Data
- ObjectMirror
- SwitchOver System

Figure 28 on page 112 illustrates the components of the DataMirror High Availability Suite. The corresponding DataMirror HA Suite Source Menu is shown in Figure 29 on page 112.

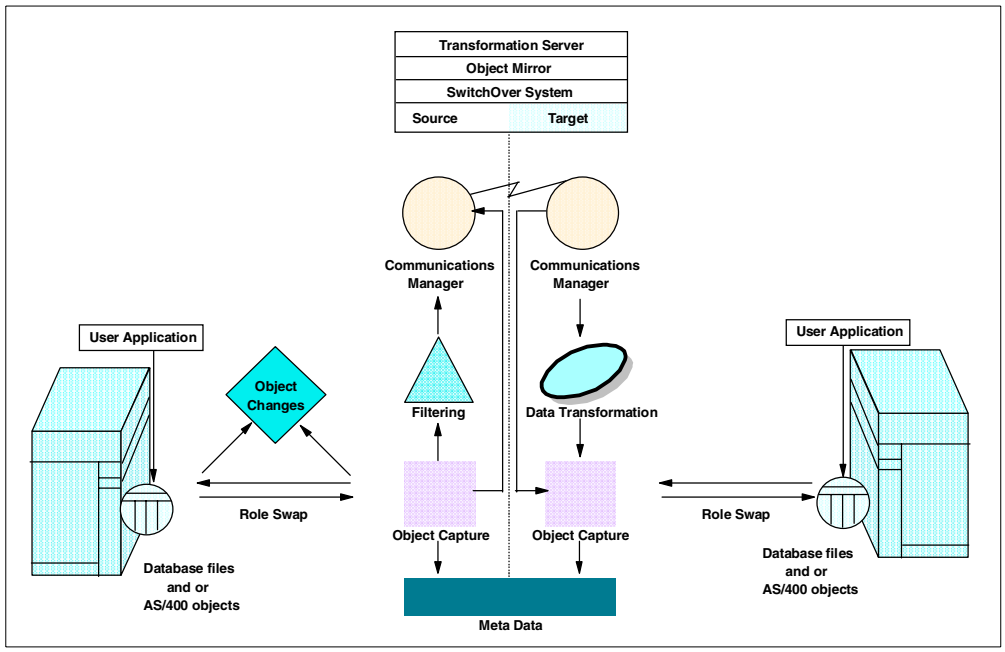


Figure 28. DataMirror High Availability Suite

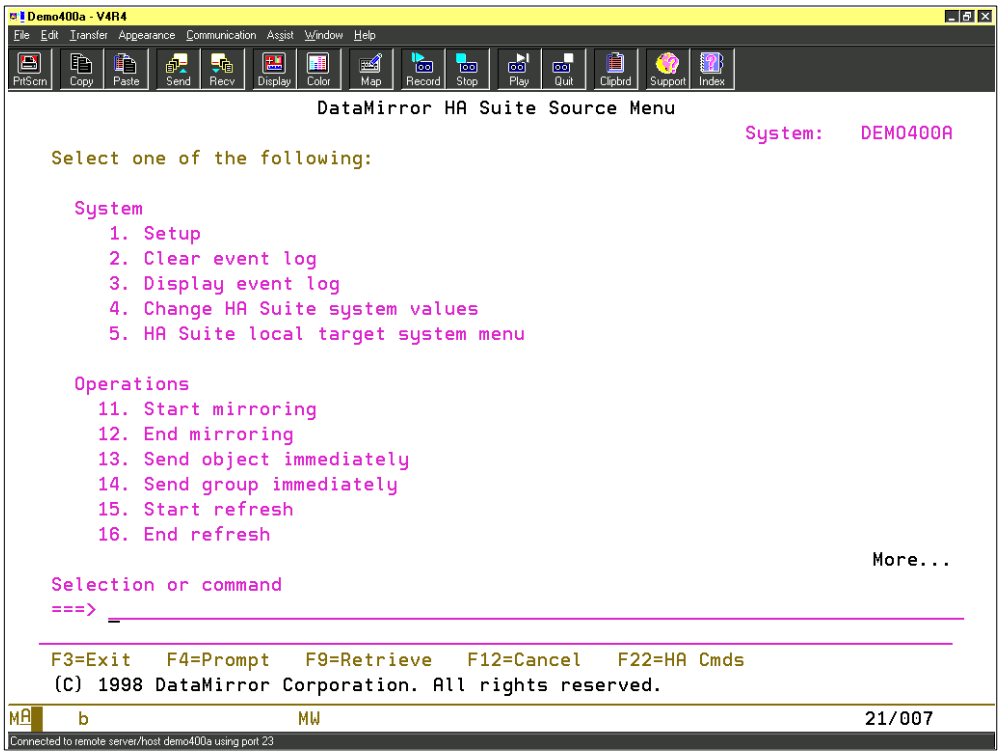


Figure 29. DataMirror HA Suite Source Menu

10.1.1 DataMirror HA Data

DataMirror HA Data mirrors data between AS/400 production systems and failover machines for backup, recovery, high systems availability, and clustering.

A user can replicate entire databases or individual files on a predetermined schedule in real time or on a net change basis. They can refresh the backup machine nightly or weekly as required. They can also use DataMirror HA Data to replicate changes to databases in real time so that up-to-the-minute data is available during a scheduled downtime or disaster.

DataMirror HA Data software is a *no-programming-required* solution. Users simply install the software, select which data to replicate to the backup system, determine a data replication method (scheduled refresh or real time), and begin replication. At the end of a system failure, fault tolerant resynchronization can occur without taking systems offline.

DataMirror HA Data supports various high availability options, including workload balancing, 7 x 24 hour operations availability, and critical data backup. Combined with Data Mirror ObjectMirror software and SwitchOver System, a full spectrum of high availability options is possible.

10.1.2 ObjectMirror

ObjectMirror enables critical application and full system redundancy to ensure access to both critical data and the applications that generate and provide data use.

ObjectMirror supports real time object mirroring from a source AS/400 system to one or more target systems. It provides continuous mirroring as well as an on-demand full refresh of AS/400 objects that are grouped by choice of replication frequency and priority.

ObjectMirror features include:

- Grouping by choice to mirror like-type objects based on frequency or priority
- Continuous real time mirroring of AS/400 objects
- Intelligent replication for guaranteed delivery to backup systems even during a system or communication failure
- Object refresh on a full-refresh basis as needed
- Fast and easy setup, including an automatic registration of objects
- Ability to send an object or group of objects immediately without going through product setup routines

10.1.3 SwitchOver System

The SwitchOver System operates on both the primary and backup AS/400 systems to monitor communications or system failures. During a failure, the SwitchOver System initiates a logical role switch of the primary and backup AS/400 systems either immediately or on a delayed basis.

A Decision Control Matrix in the SwitchOver System allows multiple line monitoring, detailed message logging, automated notification, and user-exit processing at various points during the switching process. An A/B switch (shown in Figure 30 on page 114) allows the user to automatically switch users and hardware peripherals, such as twinax terminals, printers, and remote controllers.

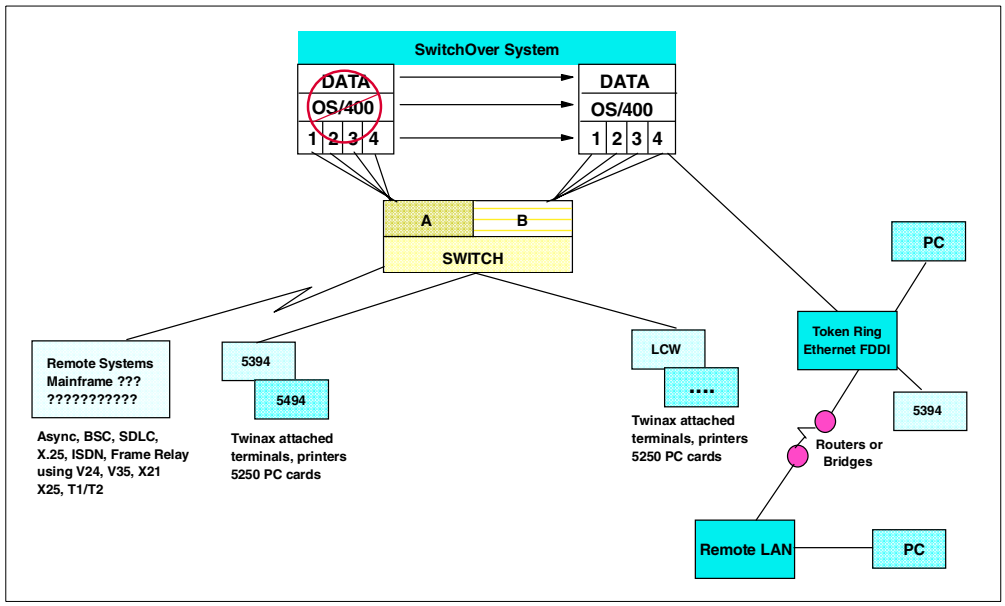


Figure 30. DataMirror SwitchOver system

10.1.4 OptiConnect and DataMirror

The DataMirror High Availability Suite supports SNA running over OptiConnect between multiple AS/400 systems. After OptiConnect is installed on both source and target AS/400 systems, the user needs to create controllers and device descriptions.

After controllers and devices are varied on, the user simply specifies the device name and remote location used in the DataMirror HA Data or Object Mirror target definition. Files or objects that are specified can then be defined, assigned to the target system, and replicated.

10.1.5 Remote journals and DataMirror

The DataMirror High Availability (HA) Suite is capable of using the IBM remote journal function in OS/400. The architecture of the DataMirror HA Suite allows the location of the journal receivers to be independent from where the production (source) or failover (target) databases reside. Therefore, journal receivers can be located on the same AS/400 system as the failover database. This allows the use of DataMirror *intra-system* replication to support remote journals.

Customers can invoke remote journal support in new implementations. Additionally, the existing setup can be modified if remote journal support was not originally planned.

10.1.6 More information about DataMirror

To learn more about DataMirror products, visit DataMirror on the Internet at:
`http://www.datamirror.com`

10.2 Lakeview Technology solutions

Lakeview Technology, an IBM business partner, offers a number of products to use in an AS/400 high availability environment. Their high availability suite contains five components:

- MIMIX/400
- MIMIX/Object
- MIMIX/Switch
- MIMIX/Monitor
- MIMIX/Promoter

The following sections highlight each of these components.

10.2.1 MIMIX/400

MIMIX/400 is the lead module in the Lakeview Technology MIMIX high availability management software suite for the IBM AS/400 system. It creates and maintains one or more exact copies of a DB2/400 database by replicating application transactions as they occur. The AS/400 system pushes the transaction data to one or more companion AS/400 systems. By doing this, a viable system with up-to-date information is always available when planned maintenance or unplanned disasters bring down the primary system. MIMIX/400 also supports intra-system database replication.

Figure 31 shows the basic principles of the MIMIX/400.

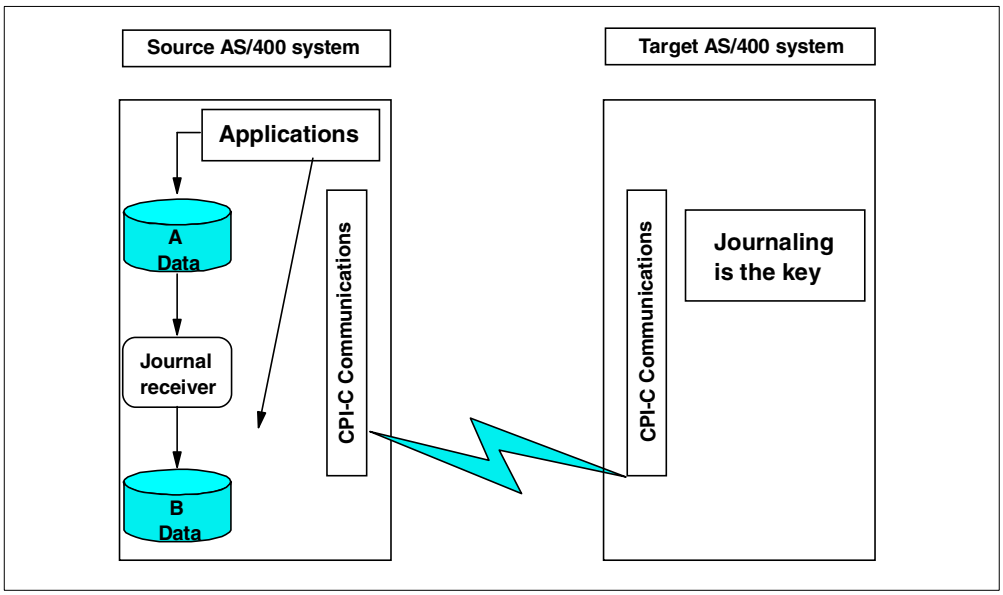


Figure 31. The basics principles of Mimix/400

The key functions of MIMIX/400 are:

- **Send**: This function scrapes the source system journal and sends the data to one or more target systems. This function offers the following characteristics:
  - Is written in ILE/C for high performance
  - Uses CPI-C to provide a low-level generic interface that keeps CPU overhead to a minimum

  - Supports filtering to eliminate files from MIMIX copies and to optimize communication throughput, auxiliary storage usage, and performance on the target system
  - Generates performance stamps, which continue throughout the replication cycle, for a historic view of performance bottlenecks
- **Receive**: This function collects transactions from the Send function. The Receive function stores and manages the transactions on the backup system for processing using the Apply function. The Receive function offers these features:
  - A temporary staging step where transactions are pushed off the sending system as soon as possible to eliminate the load from the source system CPU
  - Fast performance because it is written in ILE/C
  - Variable length log space entries to make the most of available CPU and DASD resources
  - Filtering capabilities for greater capacity exists on the target system to boost performance on the Send side
- **Apply**: This function reads all transactions and updates the duplicate databases on the target systems accordingly. The Apply function supports the following features:
  - Offers a file or member control feature to manage file name aliases and define files to the member level (files are locked during the Apply process for maximum configuration flexibility and to prevent files from being unsynchronized)
  - Opens up to 9,999 files simultaneously within a MIMIX Apply session
  - Supports record lengths up to 32 K in size
  - Manages DB2/400 commitment control boundaries during the Apply and Switch processes
  - Uses a log process to protect against data loss during a source system outage
  - Includes a graphical status report of source and target system activity. It displays the report in an easy-to-read format for operators to quickly identify MIMIX operating environment issues.
- **Synchronize**: This function verifies that the target system has recorded exact copies of the source system data. The Synchronize function supports the following features:
  - Offers keyed synchronization to keep target and source databases in synchronization with the unique “key” field in each record
  - Provides support tools to analyze and correct file synchronization errors by record
- **Switch**: This function prepares target systems for access by users during a source system outage. The Switch function performs the following tasks:
  - Defines systems, journals, fields, and data areas. The Send, Receive, and Apply sessions are linked into a logical unit called a *data group*.

- Uses a data group manager to reverse the direction of all MIMIX/400 transmissions during an outage
- Offers a journal analysis tool to identify transactions that may be incomplete after an outage

10.2.2 MIMIX/Object

The MIMIX/Object component creates and maintains duplicate images of critical AS/400 objects. Each time a user profile, device description, application program, data area, data queue, spooled file, PC file, image file, or other critical object is added, changed, moved, renamed, or deleted on an AS/400 production system, MIMIX/Object duplicates the operation on one or more backup systems.

The key elements of MIMIX/Object include:

- **Audit Journal Reader**: This element scrapes the source system security audit journal for object operations and passes them to the distribution reader. The features of the Audit Journal Reader include:
  - Management of objects within a library by object and type; document library objects (DLOs) by folder path, document name, and owner; and integrated file system objects by directory path and object name
  - Management of spooled-file queues based on their delivery destination
  - Explicit, generic (by name), and comprehensive (all) identification of library, object, DLO, and integrated file system names
  - An “include” and “exclude” flag for added naming precision
  - Integrated file system control to accommodate hierarchical directories, support long names, and provide additional support for byte stream files
- **Distribution Reader**: This element sends, receives, confirms, retries, and logs objects to history and message queues. The features of the Distribution Reader include:
  - Multi-thread asynchronous job support to efficiently handle high volumes of object operations
  - A load-leveling journal monitor to automatically detect a large back log for greater parallelism in handling requests
  - A history log to monitor successful distribution requests; offering reports by user, job, and date; and provides effective use of time for improving security control and management analysis
  - A failed request queue to provide error information, and for deleting and retrying options for ongoing object integrity and easy object resolution
  - An automatic retry feature to resubmit requests when objects are in use by another application until the object becomes available
  - Automatic management of journal receivers, history logs, and transaction logs to minimize the use of auxiliary storage
- **Send Network Object**: This element relies on the Audit Journal Reader, which interactively saves and restores any object from one system to another. It offers simplified generic distribution of objects manually or automatically through batch processing.

10.2.3 MIMIX/Switch

The MIMIX/Switch component detects system outages and initiates the MIMIX recovery process. It automatically switches users to an available system where they can continue working without losing information or productivity.

The key elements of MIMIX/Switch include:

- **Logical Switch**: This element controls the physical switch, communication and device descriptions, network attributes, APPC/APPN configurations, TCP/IP attributes, and timing of the communication switchover. The features of the Logical Switch include:
  - User exits to insert user-specified routines almost anywhere in the command stream to customize the switching process
  - A message logging feature to send status messages to multiple queues and logs for ensuring the visibility of critical information
- **Physical Switch**: This element automatically and directly communicates with the gang switch controller to create a switchover. The features of the Physical Switch include:
  - A custom interface to the gang switch controller to switch communication lines directly
  - An operator interface to facilitate manual control over the gang switch controller
  - Remote support to initiate a switch through the gang switch controller from a distance
  - Interface support of twinax, coax, RJ11, RS232, V.24, V.35., X.21, DB9, or other devices that the user can plug into a gang switch
- **Communications Monitor**: This element tracks the configuration object status to aid in automating retry and recovery. An automatic verification loop ensures that MIMIX/Switch only moves users to a backup system when a genuine source system outage occurs.

10.2.4 MIMIX/Monitor

The MIMIX/Monitor component combines a command center for the administration of monitor programs and a library of plug-in monitors so the user can track, manage, and report on AS/400 processes.

MIMIX/Monitor regulates the system 24 hours a day. It presents all monitor programs on a single screen with a uniform set of commands. This minimizes the time and effort required to insert or remove monitors or change their parameters. The MIMIX/Monitor also accepts other data monitoring tools created by customers and third-party companies into its interface.

The user can set the programs included with MIMIX/Monitor to run immediately, continually at scheduled intervals, or after a particular event (for example, a communications restart).

MIMIX/Monitor includes prepackaged monitor programs that the user can install to check the levels in an uninterrupted power supply (UPS) backup system, or to evaluate the relationship of MIMIX to the application environment.

10.2.5 MIMIX/Promoter

The MIMIX/Promoter component helps organizations maintain continuous operations while carrying out database reorganizations and application upgrades, including year 2000 date format changes. It uses data transfer technology to revise and move files to production without seriously affecting business operations.

MIMIX/Promoter builds copies of database files record-by-record, working behind the scenes while users maintain read-and-write access to their applications and data. It allows the user to fill the new file with data, change field and record lengths, and, at the same time, keep the original file online.

After copying is complete, MIMIX/Promoter moves the new files into production in a matter of moments. This is the only time when the application must be taken offline.

Implementing an upgrade also requires promoting such non-database objects as programs and display files. To handle these changes, many organizations use change management tools, some of which can be integrated with MIMIX/Promoter′ s data transfer techniques.

10.2.6 OptiConnect and MIMIX

MIMIX/400 integrates OptiConnect for OS/400 support for the IBM high-speed communications link, without requiring separate modules. The combination of MIMIX and OptiConnect provides a horizontal growth solution for interactive applications that are no longer contained on a single machine. OptiConnect delivers sufficient throughput for client/server-style database sharing among AS/400 systems within a data center for corporate use. MIMIX/400 complements the strategy by making AS/400 server data continuously available to all clients.

10.2.7 More Information About Lakeview Technology

For more information about the complete Lakeview Technology product line, visit Lakeview Technology on the Internet at: `http://www.lakeviewtech.com`

10.3 Vision Solutions: About the company

Vision Solutions was founded in 1990 by two systems programmers working at a hospital IT staff in California. They recognized the need for a dual systems solution that would exploit the rich OS/400 architecture and provide an application for managing business integrity using dual AS/400 systems. Originally known as Midrange Information Systems, Inc., the name was changed to Vision Solutions, Inc. in July 1996. Today it has grown into an international company with development staff and facilities in the Netherlands, South Africa, and the United States. It employs over 150 people worldwide.

10.3.1 Vision Solutions HAV solutions

When you consider the costs of purchasing additional assets in the form of hardware, software, and consulting services to expand the hours of operations, to increase the scope of a business' growth capability, and to allow greater utilization of a business solution on the AS/400 platform, using dual systems for mirroring the business application system is a prudent business decision. If the

focus is strictly on the disaster aspects of dual systems, the decision to go with this solution is never quick or easy to make. By expanding the view of why a business must use dual systems to the other advantages of continuous operations support, improved availability from dedicated backup processes and workload balancing, the decision process leads to a wise business project.

Vision Solutions supports this effort with its management and integrity facilities that are built directly into Vision Suite. One main advantage of using Vision Suite for your HSA and Continuous Operations requirements is that many of its application integrity features exceed the requirements of most AS/400 mission critical applications today. When considering dual systems, pursue your evaluations with due diligence and use the following criteria:

- **Integrity**: How do you know the backup system is equal to your production? Do you employ more analysts to write additional utilities for monitoring or support that use your existing staff? Or, will this decision software reside in the HSA solution?
- **Performance**: How much data can you push to the other system to minimize in-flight transactions being lost during an unplanned outage using the minimum possible CPU? If your application employs OS/400 Remote Journaling and Clustering, does the HSA vendor demonstrate live support of this capability?
- **Performance**: What happens when your network stops or your backup system fails for an indefinite period of time? Can you catch up quickly to protect your business?
- **Ease of Use**: Can my existing operations staff use this application?
- **Application Support**: As an application evolves and possibly extends its use of the rich OS/400 architecture, will your HSA application be able to support those extensions without customized software?

Pursue this criteria for your business needs and commitments with regards to continuous operations and business integrity.

10.3.2 Vision Suite

There are three components to Vision Suite:

- Object Mirroring System/400 (OMS/400)
- Object Distribution System/400 (ODS/400)
- System Availability Monitor/400 (SAM/400)

The requirements for Vision Suite necessitate the use of the OS/400 Journal function for both OMS/400 and ODS/400. Journaling gives Vision Suite the ability to deliver real time database transactions and event driven object manipulations to a backup AS/400 system. While some AS/400 application environments have journaling already active, the integrated Vision Suite Journal Manager can relieve the user of the journal receiver management function. Figure 32 on page 121 illustrates a typical configuration between a production system and a backup system. The journal receivers provide the input to Vision Suite for replication to the backup Database Server system.

10.3.2.1 OMS/400

This component replicates and preserves the Application Integrity established in your software design. It immediately transfers the transitional changes that occur

in your data areas, data queues, and physical files to a backup AS/400 system. As the application database is manipulated by the programs I/O requests, and these operations are recorded in the journal by OS/400, OMS/400 transports the resulting journal entries of those requests to a backup database server in real time. This minimizes any data loss due to an unplanned outage. As shown in Figure 32, the Reader/Sender function takes the journal entry over to the backup system through various communications media supported in OS/400. Any communications media utilizing the SNA or TCP/IP protocols and OptiMover are supported by Vision Suite.

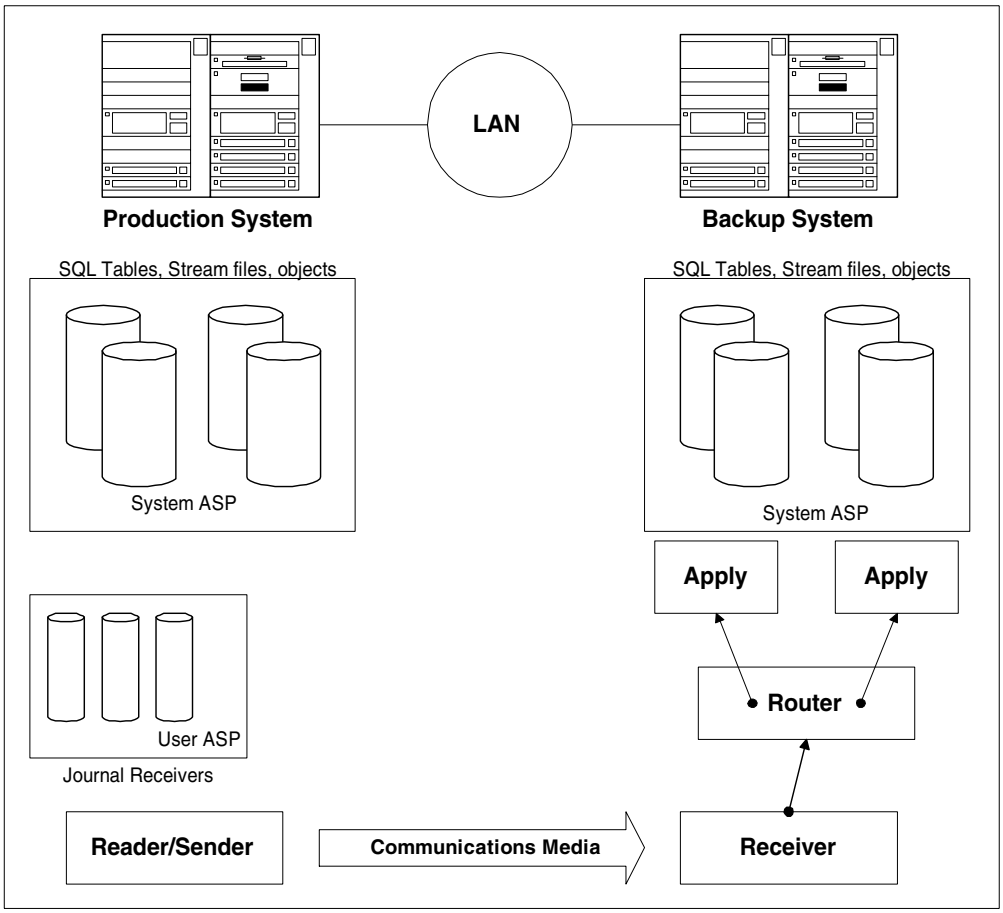


Figure 32. Typical Vision Suite configuration

The Receiver function places the journal entry into the Router function so that it may be assigned to an Apply Queue. The Apply Queue writes the record image captured in the journal entry into the appropriate data object located on the backup system. All objects in a given Database Server are distributed evenly across multiple Apply Queues provided for that single database. The equal separation of files (according to logical relationships in DB2/400 or applications) ensures an even use of CPU and memory resources among each Apply Queue.

10.3.2.2 Application integrity

In addition to the requirement of delivering data efficiently and quickly, OMS/400 manages the synchronization and integrity of the database and data objects by using several background processes that do not require operator control and management. Integrity of your database between each physical file, data area,

and data queue, along with the applications, is a cornerstone for successful role swaps. Role swap is Vision terminology for moving the business processes, which can be end users, batch jobs, or a combination of both application environments from one AS/400 server to another.

Assurances for complete Application integrity on your backup systems allow you to quickly declare disasters instead of hoping for something else to happen. No integrity issue of any type ensures that planned or unplanned outages occur with a quick role swap supporting continuous operations at minimum downtime.

Furthermore, the journal receivers generated from all of the database activity impacts the auxiliary storage space in a short time if these objects are not managed. Vision Suite features its Journal Manager function that creates journal environments for your mission critical databases. It can completely control the entire management role of journal receivers on both the production and backup AS/400 server. This ensures that the client is free for other AS/400 maintenance functions.

10.3.2.3 ODS/400

The second component of Vision Suite preserves the Environment Integrity developed for your application by replicating all supported object types of that application. This is an important consideration for any AS/400 system requiring continuous operations. While database transactions are complex and numerous compared to object manipulations, change management of your application environment must be duplicated on the backup system to ensure a timely and smooth role swap (move the business from the production system to the backup system). To ensure Environment Integrity, a user should choose to perform change management on each individual system or use ODS/400 to replicate the various object changes to those systems.

Increasingly, *object security* has heightened the need for ODS/400. In pre-client (or PC Support) days, typical security control was managed through 5250 session menus. However, today's WAN and Internet/intranet network environments utilize many application tools that are built on ODBC and OLE database interfaces. AS/400 IT staffs must meet the challenge of taking advantage of OS/400 built-in object security. This involves removing public authority from mission critical objects and interjecting the use of authorization lists and group profiles. While the AS/400 system maintains a high-level interface for this work, the interlocking relationships of objects, databases, and users (both local and remote) become complex. ODS/400 maintains this complex environment so that its integrity is preserved on your backup system.

In mission critical AS/400 applications, the main focus for continuous operations and high availability is the integration of the database with its associated software and related security object authorizations and accesses.

10.3.2.4 SAM/400

The final component of Vision Suite is SAM/400. Its main purpose is to monitor the production (or source application) system heartbeat and condition the role swap when all contact is lost. It has ancillary functions for keeping unwarranted users from accessing the backup system when it is not performing the production function. It also provides user exits for recovery programs designed by the Professional Services staff for specific recovery and environment requirements.

Vision Solutions, Inc. products operate on two or more AS/400 systems in a network and use mirroring techniques. This ensures that databases, applications, user profiles, and other objects are automatically updated on the backup machines. In the event of a system failure, end users and network connections are automatically transferred to a predefined backup system. The Visions products automatically activate the backup system (perform a role swap) without any operator intervention.

With this solution, two or more AS/400 systems can share the workload. For example, it can direct end-user queries that do not update databases to the backup system. Dedicated system maintenance projects are another solution benefit. The user can temporarily move their operations to the backup machine and upgrade or change the primary machine.

This High Availability Solution offers an easy and structured way to keep AS/400 business applications and data available 24 hours a day, 7 days a week.

The Vision Solutions, Inc. High Availability Solution (called Vision Suite) includes three components:

- Object Mirroring System (OMS/400)
- Object Distribution System (ODS/400)
- System Availability Monitor (SAM/400)

The following sections highlight each component.

10.3.3 OMS/400: Object Mirroring System

The Object Mirroring System (OMS/400) automatically maintains duplicate databases across two or more AS/400 systems.

Figure 33 illustrates the OMS/400 system. This system uses journals and a communication link between the source and target systems.

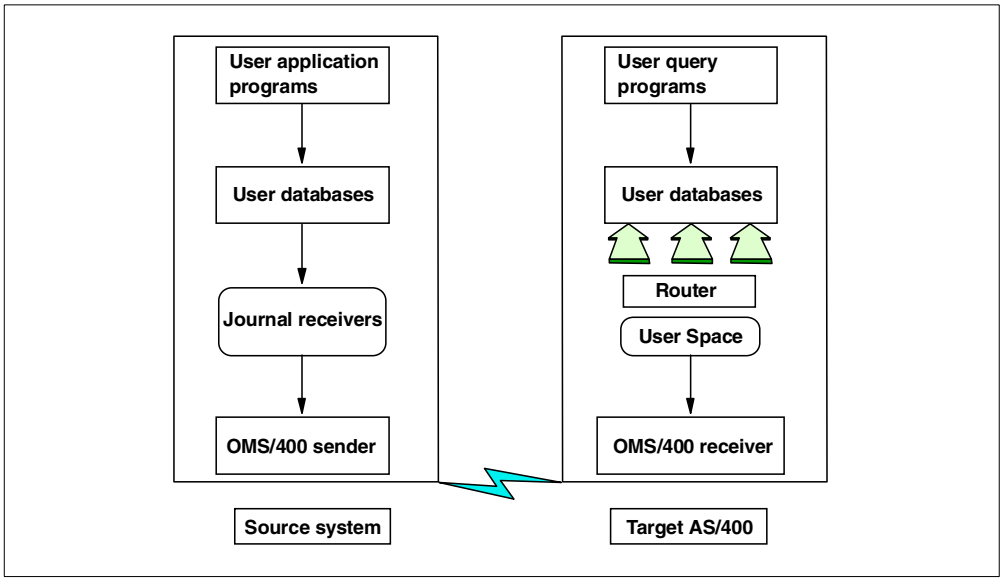


Figure 33. Object Mirroring System/400

The features of the OMS/400 component include:

- Automatic repair of such abnormal conditions as communication, synchronization, or system failure recoveries
- Synchronization of enterprise-wide data by simulcasting data from a source system to more than 9,000 target destinations
- User space technology that streamlines the replication process
- An optional ongoing validity check to ensure data integrity
- Automatic restart after any system termination
- Automatic filtration of unwanted entries, such as opens and closes
- The power to operate programs or commands from a remote system
- The ability to dynamically capture data and object changes on the source system and copy them to the target system without custom commands or recompiles
- The option to create an unlimited number of prioritized AS/400 links between systems
- Total data protection by writing download transactions to tape
- Support of RPG/400 for user presentation, and ILE/C for system access, data transmission, and process application
- Full support of the IBM OptiConnect/400 system
- Global journal management when a fiber optic bus-to-bus connection is available
- The use of CPI-C to increase speed of data distribution using minimal CPU resources

10.3.4 ODS/400: Object Distribution System

The Object Distribution System (ODS/400) provides automatic distribution of application software, authority changes, folders and documents, user-profile changes, and system values. It also distributes subsystem descriptions, job descriptions, logical files, and output queue and job queue descriptions.

ODS/400 is a partner to the OMS/400 system, and it provides companies with full system redundancy. It automatically distributes application software changes, system configurations, folders and documents, and user profiles throughout a network of AS/400 computers.

ODS/400 supports multi-directional and network environments in centralized or remote locations. For maximum throughput, ODS/400 takes advantage of bi-directional communications protocol and uses extensive filters.

10.3.5 SAM/400: System Availability Monitor

The System Availability Monitor (SAM/400) can switch users from a failed primary system to their designated secondary system without operator intervention.

SAM/400 works in conjunction with OMS/400 and ODS/400, continuously monitoring the source system. In the event of a failure, SAM/400 automatically redirects users to the target system. This virtually eliminates downtime. High-speed communications links, optional electronic switching hardware, and

SAM/400 work together to switch users to a recovery system in only a few minutes.

SAM/400 offers:

- Continuous monitoring of all mirrored systems for operational status and ongoing availability
- A fully programmable response to react automatically during a system failure, which reduces the dependence on uninformed or untrained staff
- The ability to immediately and safely switch to the target system, which contains an exact duplicate of the source objects and data during a source system failure (unattended systems are automatically protected 24 hours a day, 7 days a week)
- User-defined access to the target system based on a specific user class or customized access levels

The SAM/400 component offers:

- Up to ten alternate communication links for monitoring from the target system to the source system
- Automatic initiation of user-defined actions when a primary system failure occurs
- Exit programs to allow the operator to customize recovery and operations for all network protocols and implementations

Figure 34 illustrates the SAM/400 monitoring process.

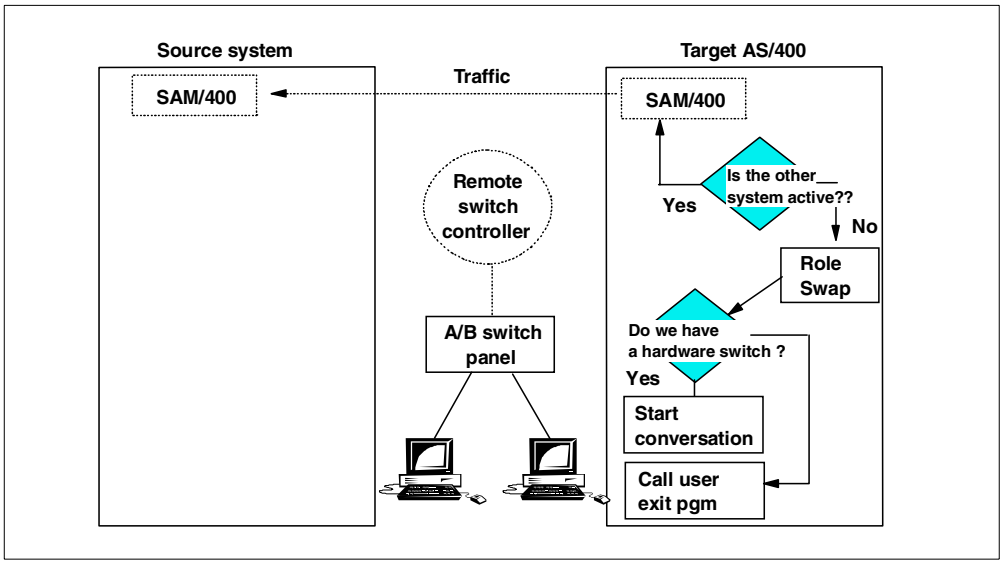


Figure 34. SAM/400 structure

Users are allowed to access applications at the “End” point.

10.3.6 High Availability Services/400

High Availability Services/400 (HAS/400) consists of software and services. The HAS/400 solution is comprised of:

- Analysis of the customer environment in terms of system availability needs and expectations, critical business applications, databases, and workload distribution capabilities
- An implementation plan written in terms of solution design and the required resources for its deployment
- Installation and configuration of the software products and the required hardware
- Education for the customer staff on operational procedures
- Solution implementation test and validation
- Software from Vision Solutions, Inc., as previously described

10.3.7 More information about Vision Solutions, Inc.

For more information about Vision Solutions, Inc. products, visit Vision Solutions on the Internet at: `http://www.visionsolutions.com`

Chapter 11. Application design and considerations

Applications are regarded as business-critical elements. The viewpoint of systems management is changing from a component view to an application view. Here are a few considerations that are now made at the application level:

- The entire application, or parts of the application, must be distributed.
- The application has to be monitored to guarantee availability.
- Operations, such as scheduling jobs and doing backups, are recommended.
- User profiles must be created, given access to applications, changed, and deleted.

If a system is unavailable, and rapid recovery is necessary, backups are restored and the system and the database are inspected for integrity. The recovery process could take days for larger databases.

This scenario requires improvement in the areas of restore speed and application recovery. Both of these areas can be costly to implement. High speed tape drives are very expensive items and, for very large databases, they may not show enough restore time improvement to meet user demands.

Application recovery requires a lot of development effort and is, therefore, very costly. This, in itself, may also degrade availability. To make the application more available, considerable additional processing and I/O should be available. This means the response time degrades unless there is an abundance of computing resources.

In the past, application recovery was at the bottom of the list of availability tasks. The size of required systems would be too expensive for the business to justify. These days, with the vast improvements on price and performance power, solutions exist that provide a high level of availability. In addition, businesses are now able to define the cost of system outage more accurately.

From a user point-of-view (both a system end user, and a receiver of services), the availability of a system relates to the information available at a given time. A customer holding for a price lookup considers the efficiency of the answer as an indication of the quality of the business.

Designing applications for high availability is a comprehensive topic and textbooks have been dedicated to this topic alone. From a high level, some of the considerations are discussed in this chapter. Areas that are covered include application checkpointing design, considerations, and techniques (including CL programs), for the interactive environment.

11.1 Application coding for commitment control

You can use commitment control to design an application so that it can be started again if a job, an activation group within a job, or the system ends abnormally. The application can be started again with the assurance that no partial updates are in the database due to incomplete logical units of work from a prior failure.

There are numerous documents that describe the use of commitment control and journaling. The *OS/400 Backup and Recovery,* SC41-5304, contains journaling and commitment control requirements. IBM language-specific manuals include:

- *DB2 for AS/400 SQL Programming, Version 4,* SC41-5611
- *ILE C for AS/400 Programmer's Guide, Version 4,* SC09-2712
- *ILE COBOL for AS/400 Programmer's Guide, Version 4,* SC09-2540
- *ILE RPG for AS/400 Programmer's Guide, Version 4* SC09-2507

These manuals contain information about using commitment control for a particular language. Various Redbooks and "how to" articles are found throughout IBM-related web sites. These sites include:

- `http://www.news400.com`

  "Safeguard Your Data with RI and Triggers", Teresa Kan December, 1994 (page 55)

- `http://www.news400.com`

  "*AS/400 Data Protection Methods"*, Robert Kleckner, December 1993 (page 101)

11.2 Application checkpointing

In general, application checkpointing is a method used to track completed job steps and pick up where the job last left off before a system or application failure. Using application checkpointing logic, along with commitment control, you can provide a higher lever of resiliency in both applications and data, regardless of whether they are mission critical in nature.

Throughout the existence of IBM midrange systems, application checkpointing has been used to help recover from system or application failures. It is not a new subject when it comes to IBM midrange computing, specifically on the AS/400 system. However, there are some new features.

Unlike commitment control, application checkpointing has no system level functions that can be used to automate recovery of an application. If commitment control is used, and the job stream has multiple job steps, the application needs to know which job has already run to completion. Application checkpoints help the programmers to design recovery methods that can prevent the restarted jobs from damaging the database by writing duplicate records.

Remember that there is additional information written on concepts and methods for application checkpoints, as well as recovery with or without journaling and commitment control.

The following sections define recovery methods for applications that work in any high availability environment for the AS/400 system (including clustering). Most of the concepts also work for other (non-AS/400) platforms.

11.3 Application checkpoint techniques

Techniques for application checkpointing and recovery vary for every program. Whether you use Cobol, RPG, SQL or C as the application language, the methods employed for application checkpointing remain constant.

Without journaling and commitment control capabilities, programmers devise their own tracking and recovery programs. This section describes an example of how this is done.

11.3.1 Historical example

The following scenario describes a customer environment that runs the sales force from a System/36 in the early 1980s.

The customer's remote sales representatives dial into a BBS bulletin board system from their home computer, upload the day's orders, and request the sales history for the clients they are to visit the next day. The next morning, the same remote sales representative dials into the BBS to retrieve the requested information.

BBS systems and modems were not reliable in the early 1980s. Many of the transfers ended abnormally. Application programmers devised a method to update data areas with specific job step information after the job steps completed. If the Operator Control Language (OCL) job starts and finds an error code in the data area, the program logic jumps to the last completed step (as indicated by the data area) and starts from there. This is a primitive form of application checkpointing, but it works.

Later applications utilized log files. Programs were designed to retrieve information from the log file if the last job step was not successful. Using the program name, last completed job step, current and next job step, as well as the total job steps, the programmer determined where to start the program. The program itself contained recovery subroutines to process if the recovery data area contained information that a job failure occurred.

As for the data, temporary files were created containing the before image of a file. Reading the required record, then writing the temporary file prior to any updates or deletes, the information was written back to its original image if the job failed. At the completion of the job, all temporary files were removed.

With the high availability products on the market today, a more efficient design is possible. A permanent data file includes the data area logic. High availability products mirror data areas and data queues. However, the HA applications work off of journal information.

Note: Data areas and data queues can not be journaled in OS/400 V4R4. Moving any checkpoint logic from a data area to a data file operates with more efficiency and provides a higher restart capability than data areas.

11.4 Application scenarios

The following paragraphs explain application checkpointing methods in various scenarios. The methods described are not the only possible options for application checkpoints. However, they provide a good starting point for managing your high availability environment with application checkpoints that work in any HA environment.

11.4.1 Single application

For a single application, checkpoints are established by adding recovery logic to the program to handle the commit and roll back functions. The job's Control Language (CL) program needs to include checks for messages that indicate an incomplete or open Logical Unit of Work (LUW).

Testing for incomplete job runs is the primary requirement of application checkpoints. Some simple testing of control information for an error code prior to running the start or end commit command prevents users from getting erroneous messages. If the control information is clean, run the Start Commitment Control (STRCMTCTL) function and change the control information to *uncompleted*. If the control information has an error code, act on it by performing either a commit or rollback. Most often, the action is a rollback. At the completion of the program, execute a commit and then change the control information to indicate a successful update.

11.4.2 CL program example

This example uses CL programs. It is assumed that the Logical Unit of Work (LUW) includes all I/O operations that this program performs. If a rollback takes place, all changes are removed from the system.

If there is a higher complexity to the application, such as multiple levels of application calls, or many updates, inserts, and deletes, you should consider this a multiple application program.

Also, in this example, a data area is used for the control information. For High Availability, it is recommended that you have the control information in a data file. Mirroring record information is more efficient than data areas or data queues because the current release of OS/400 (V4R4) does not support journaling data areas or data queues. Successful and complete recovery from a system failure is more likely if the recovery information is contained in a mirrored file.

If this job is used in a multi-step job stream, place true application checkpoint functions into it. To do this, create a checkpoint-tracking file. The checkpoint-tracking file used to track the job steps must include information about the job and where to start.

```
PGM
DCL &OK *CHAR 1
RTVDTAARA CONTROL &OK 1 1
IF COND(&OK *NE ' ') THEN(ROLLBACK)
CHGDTAARA &OK 'E'
STRCMTCTL
CALL UPDPGM
RTVDTAARA CONTROL &OK 1 1
IF COND(&OK *NE ' ') THEN(ROLLBACK)
IF COND(&OK *EQ ' ') THEN(COMMIT)
ENDCMTCTL
CHGDTAARA &OK ' '
ENDPGM
```

Chapter 12. Basic CL program model

The following model contains the basic information for most jobs. Additional information can and should be tracked for better recovery:

```
* Program Name
* Current Job Step
* Previous Job Step
* Next Job Step
* Total Job Steps
* Job Start time
* Job Name
* User
* Last processed record key information
```

The information listed here should help you determine where you are, where you were, and where to go next. You can also determine how far into the job you are and who should be notified that the job was interrupted.

12.1 Determining a job step

The diagrams shown in Figure 35 and Figure 36 illustrate how to determine a job step.

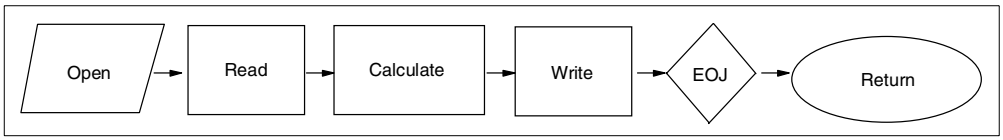


Figure 35. Determining a job step (Part 1 of 3))

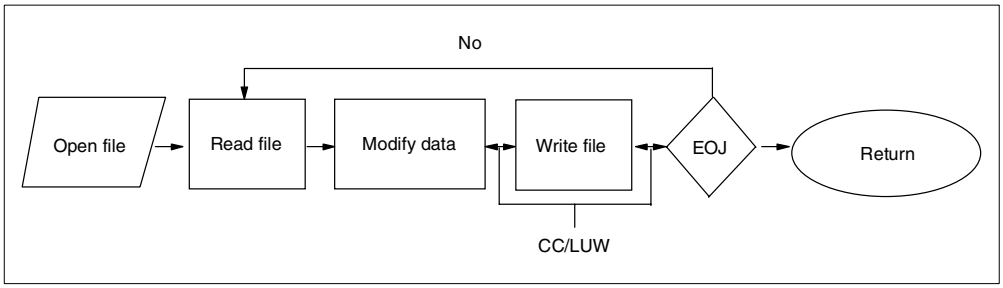


Figure 36. Determining a job step (Part 2 of 3)

All programs have some basic flow. Using Commitment Control, the data is protected with the LUW, Commit, and Rollback. The diagram shown in Figure 37 on page 132 shows these components.

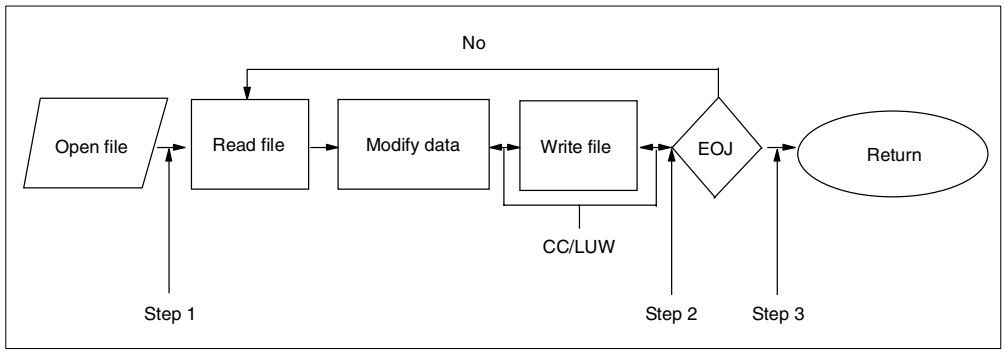


Figure 37. Determining a job step (Part 3 of 3)

To determine the key points for the job step markers, notice how the data is read, manipulated, and written. Also look at any end-of-job processing.

If the job fails while processing the data, start at the last completed processed record. This means that the section where the data is read into the program is a “key point” for your job step. Most applications read the data first, so this is the first job step. If there is a section of the program prior to the read that must be recalculated, it should be the first point of the job step.

After determining where the data is read, look at where the data is written. In this example, commitment control provides the second key point. When the data is committed to a disk, it can’t be changed. By placing the next job step at this point, a restart bypasses any completed database changes and moves to the next portion of the program.

If the program writes thousands of records, but performs a commit at every 100, the checkpoint tracking information should include some key elements for the last committed record. This information should be collected for step 1 after every commit. With the information collected this way, a restart can go to step 1 and set the initial key value to start reading at the last committed point.

End of job processing can cause many application restart attempts to fail. The reasons for this should appear obvious. If the job does not have an end of job (EOJ) summary or total calculations to perform, then step 2 is the last job step. However, if summary reports and total calculations must be performed (for example, for an invoice application), some added logic is needed. Most summary or total calculations are performed on data that is collected and calculated from a previously performed file I/O. They are usually stored in a table or in an array in memory. If the job fails before EOJ calculations are made, memory allocated for the job is released. Take this into consideration when working with commitment control’s logical unit of work. The logic for determining the recovery job steps may need to be “Start everything over”. With improper commitment control logic, data loss is possible.

If, on the other hand, the job is extremely long running or is a critical job, write the summary information into a work file. The process of completing a job step and writing out the work file occurs before EOJ processing and can be considered its own job step. If you use a work file for array or table information, place the name of the work file in the tracking file as well. Include restart logic in the program to reload any tables or arrays for EOJ processing.

Keep in mind that high availability solution providers can track and mirror files that are created and deleted on the fly. However, processing overhead is much higher and the chance of “in-flight transactions” can occur at the time of failure. This results in lost data. Therefore, it is recommended that work files be created and maintained instead of created and deleted.

Using the Clear Physical File Member (CLRPFM) function on the AS/400 system maintains the journal link with the file and reduces the overhead of saving and restoring file information caused by a create and delete. Create temporary files in the QTEMP library on the AS/400 system. QTEMP can not be mirrored by high availability solution providers. This library can contain true temporary files in which the contents can be easily recreated. If a job fails, temporary files do not affect the program outcome.

The last checkpoint in the job should take place immediately before the end of, or return to, the previous job step. This checkpoint can clear any error flags, reset the program name, or remove the tracking record from the checkpoint tracking file. Creating logging functions for history reporting can also be performed from this checkpoint.

12.1.1 Summary of the basic program architecture

The examples and logic defined in 11.3, “Application checkpoint techniques” on page 128, work within the AS/400 system on all release levels of OS/400 from Version 1 onward. If you plan on moving to a clustered environment available from IBM (from OS/400 V4R4 onward), the model described here provides a good start for cluster ready.

Add additional tracking information to start the users at a specific point in the application, such as tracking screens and cursor positions. When writing out the checkpoint-tracking file, reserve space for this information so the recovery process can place the user as close to where they left off as possible. Information for screens and cursor positions, as well as job information, user information, and file information can be retrieved from the Information Data Structures (INFDS) provided by OS/400.

A wealth of information can be retrieved from the INFDS, including screen names, cursor positions, indicator maps, job information, and more. Part of the cluster ready requirements is to restart the users where they last left off. Using application checkpoints accomplishes this.

12.2 Database

At the core of the AS/400e system is the relational database. When multiple systems are involved, a portion or a copy of the data can reside on the remote system. This is called a distributed database.

12.2.1 Distributed relational database

Application coding for distributed databases require more involved and complicated logic than programs designed for accessing data on a single system. Application checkpoint logic adds further complexity to the program logic. Both of these situations require a seasoned programmer analyst. This should be someone with patience, persistence, and experience. This section addresses

some of the concerns of programming in a distributed relational database environment.

Note: Using the ideas and concepts discussed in this chapter, and deploying them into any distributed application model, does not change the amount of work for the programmer. Application checkpointing is not concerned with where the data is. Rather, it is concerned with where the application runs.

If you use DRDA as the basis of the distributed database, the applications reside with the data on all systems. DRDA submits a Unit of Work (UoW) that executes a sequence of database requests on the remote system. In other words, it calls a program and returns the information.

DRDA uses a two-phase commit for the database. This means that the data is protected from failure over the communications line, as well as system or job failures.

If the program on the remote system performs proper application checkpointing when the remote system, remote application, or communications, fails during the processing of UoW, the restart picks up from where it left off.

Adding a “from location” field and “to location” field in the checkpoint-tracking file allows reports information that better defines the locations of the jobs running. It also helps isolate communication fault issues with the application modules that start and stop communications.

It is recommended that application checkpointing in a DRDA environment be setup in a modular fashion. A recovery module that controls the reads and writes to the checkpoint-tracking file make analysis and recovery easier. Take special care to ensure that the database designers include the checkpoint-tracking file in the original database design. This enables the recovery module to treat each existence of the tracking file as an independent log for that local system.

A future release of DB2/400 UDB Extended Enterprise Edition (EEE) will include scalability for Very Large Database (VLDB) support. Like DRDA, the VLDB database is distributed over different systems. Unlike DRDA, the access to the data is transparent to the program. No special communication modules or requester jobs need to be created and maintained. Since the environment looks and feels like a local database, the application checkpointing logic must treat it like a local database running in the multiple application mode.

For more details on the workings of DB2/400 and DRDA, refer to *DB2/400 Advanced Database Functions*, SG24-4249.

12.2.2 Distributed database and DDM

You can use DDM files to access data files on different systems as a method of having a distributed database. When the DDM file is created, a “shell” of the physical file resides on the local system. The shell is a pointer to the data file on the remote system.

The program that reads from, and writes to, this file does not know the data is located on a different system. Since DDM files are transparent to the application, approach application checkpointing logic as discussed in 11.4.1, “Single application” on page 129.

12.3 Interactive jobs and user recovery

The information and logic necessary to recover from a system or application failure is no different between interactive jobs and batch jobs. There is a difference in how much user recovery is needed.

There are three basic parts to every job:

- Data
- Programs
- Users

Use commitment control and journaling to address data recovery.

The file layout for the checkpoint-tracking file described previously has a space reserved for the user name. This user name field can be used to inform the user that the job was abnormally ended and that a restart or recovery process will run. If this were an interactive job with screen information, the chance of the user getting back to where they left off is not high. To correct this deficit, the recovery process tracks more information to place the user as close to the point they left off as possible. The addition of current screen information, cursor positions, array information, table pointers, and variables can be stored in the tracking file along with all the other information.

The AS/400 system stores all of the information it needs to run the application accessible by the programmer. The programmer simply needs to know where to find it. The Information Data Structure (INFDS) in RPG contains most, if not all, of the information required to get the user back to the screen from where they left off.

Use the Retrieve Job Attribute (RTVJOBA) command to retrieve job attributes within the CL of the program. Retrieve system values pertinent to the job with the Retrieve System Value (RTVSYSVAL) command.

Internal program variables are in control of the application programmer. Recovery logic within the application can retrieve the screen and cursor positions, the run attributes, and system values and write them to the tracking file along with key array, table, and variable information. In the event of a failure, the recovery logic in the program determines the screen the user was on, what files were open, what the variable, array, and key values were, and even place the cursor back to the last used position.

12.4 Batch jobs and user recovery and special considerations

Unlike interactive jobs, batch jobs have no user interface that needs to be tracked and recovered. Since the user interface is not a concern, the checkpoint tracking process defined in 11.4.1, “Single application” on page 129, and 12.2.1, “Distributed relational database” on page 133, should suffice.

In general, this is true. This section describes additional vital information about when the recovery environment includes high availability providers software. These considerations include:

- **Job queue information for the batch jobs cannot be mirrored**: This means that if you submit multiple related jobs to a single threaded batch queue, and the system fails before all those jobs are completed, restarting may not help.
- **Determining what jobs have been completed and what jobs still need to run**: If you have created a checkpointing methodology with the points described previously, you have a tracking record of the job that was running at the time of the failure. Using this information, restart that job and then manually submit the remaining jobs to batch.

  If this was a day-end process, determining the jobs that still need to be run should not be complicated. If it was a month-end process, the work to restart all the jobs consumes more time but it can be achieved with few or no errors. If this was a year-end process, the work to restart all the necessary jobs in the correct order without missing vital information can be very time consuming.

  A simple solution is to store tracking information for batch jobs in the application checkpoint-tracking file. The added work in the recovery file is minimal, yet the benefits are exponential.

  Within the job that submits the batch process, a call to the application checkpointing job name, submitted time, submitted queue, and submitted status is ideal. The application checkpoint module writes this information to the checkpoint-tracking file. When the job runs in batch mode, it modifies the status to “Active”. Upon completion, remove the record or mark as was done in the tracking file to complete a log of submitted jobs.

  Within any high availability environment, the tracking information is processed almost immediately. In the event of a system failure, the recovery module interrogates the tracking file for submitted jobs that are still in a JOBQ status and automatically resubmit them to the proper job queue in the proper order (starting with the job with an “Active” status). This possibility prevents any user intervention, therefore eliminating “user error”.

12.5 Server jobs

The nature of a server job is very robust. To maintain reliability, server jobs should be able to recover from most, if not all, error conditions that can cause normal jobs to fail.

Since server jobs run in a batch environment, the recovery process itself is identical to the batch process described in 12.4, “Batch jobs and user recovery and special considerations” on page 135. However, additional considerations for error recovery are necessary for the recovery file.

With application checkpointing built into the server job, error conditions can be logged in the tracking file and corrections to either the server job or the client jobs can be made based on what is logged.

Using application checkpoints to isolate faults and troubleshoot error conditions is an added advantage to a well-designed recovery process. If the server job fails, connection to the client can still exist. If this open connection is not possible, there may be a way to notify the client to re-send the requested information or Unit-of-Work (UoW). Either way, the server job must track the information in the checkpoint-tracking file.

12.6 Client Server jobs and user recovery

Client Server jobs come in many different models, including thin clients, fat clients, and other clients that include attributes of both fat and thin clients.

Even though they have a different label, these client server jobs are either a batch job or an interactive job. The environment that the job runs in dictates the type of recovery to perform.

Most client server applications rely on the client to contact the server to request information from the server. Thin clients contact the server and pass units of work for the server to perform and report back. Fat clients request data and process the information themselves. “Medium” clients perform various aspects of each method.

Recovery for the client server jobs should be mutually exclusive. If the client job fails, the connection to the server can still exist. The client may be able to pick up where it left off.

If this open connection is not possible, there may be a way to notify the server to re-send the request for information. Either way, the client job must track the information in the checkpoint-tracking file.

If the processing of the client request pertains to a long running process, it may be best to design that particular job as a thin client. With a thin client design, the processing is performed on the server side where application checkpointing tracks and reports the job steps. In this case, recovery on the client includes checking whether the communications is still available. If not, then submit the request again from the beginning.

If the processing of the client pertains to critical information, the design should lean towards the fat client model. If the client is a fat client model, the application checkpointing logic described in this book should suffice.

Note: The nature of a client server relationship varies greatly. It is worth the time to determine whether the recovery process in a client server environment is necessary prior to writing the recovery steps. Thin clients perform much faster in a restart mode if they are simply started again with absolutely no recovery logic. For example, if a client process makes one request to the server for information, adding recovery logic can double or even triple the amount of time required to make that request.

12.7 Print job recovery

In the standard program model, information is collected, processed, and written. The process of writing the information typically occurs during the end of job (EOJ), after all the data is collected. In this case, adding a checkpoint at the beginning of the EOJ processing restarts the printing functions in the event of a restart. The scenario is described in “Distributed relational database” on page 133.

Exceptions to this rule include programs that collect and write “detailed” information as the job runs. Again, 12.2.1, “Distributed relational database” on

page 133, describes how a job step defined at the proper locations recreates every function within the steps.

Even with a detailed and proper running recovery function in place, there is no way to “pick up where you left off” in a print job. The print file itself is closed when the job runs. Rewriting to it is not possible.

With proper application checkpoints in place, the print information should not be lost. It is, however, duplicated to some extent.

If a print job within that application requires a specific name, that name should be tracked in the checkpoint-tracking file and proper cleanup should be performed prior to the job running again.

Part 4. High availability checkpoints

Part IV discusses miscellaneous items that are helpful when implementing a high availability solution. Included in this part is information on a Batch Caching solution, a discussion of the management of disk storage, device parity features, and a checklist of items to consider when implementing your high availability solution.

Appendix A. How your system manages auxiliary storage

For many businesses, computers have replaced file cabinets. Information critical to a business is stored on disks in one or more computer systems. To protect information assets on your AS/400 system, you need a basic understanding of how it manages disk storage.

On the AS/400 system, main memory is referred to as *main storage*. Disk storage is called *auxiliary storage*. Disk storage may also be referred to as *DASD* (direct access storage device).

Many other computer systems require you to take responsibility for how information is stored on disks. When you create a new file, you must tell the system where to put the file and how big to make it. You must balance files across different disk units to provide good system performance. If you discover later that a file needs to be larger, you need to copy it to a location on the disk that has enough space for the new larger file. You may need to move other files to maintain system performance.

The AS/400 system is responsible for managing the information in auxiliary storage. When you create a file, you estimate how many records it should have. The system places the file in the location most conducive for good performance. In fact, it may spread the data in the file across multiple disk units.

When you add more records to the file, the system assigns additional space on one or more disk units. The system uses a function that is called virtual storage to create a logical picture of how the data looks. This logical picture is similar to how data is perceived. In virtual storage, all of the records that are in a file are together (contiguous), even though they may be physically spread across multiple disk units in auxiliary storage. The virtual storage function also keeps track of where the most current copy of any piece of information is (whether it is in main storage or in auxiliary storage).

Single-level storage is a unique architecture of the AS/400 system that allows main storage, auxiliary storage, and virtual storage to work together accurately and efficiently. With single-level storage, programs and system users request data by name rather than by where the data is located.

Disk storage architecture and management tools are further described in *AS/400 Disk Storage Topics and Tools,* SG24-5693.

A.1 How disks are configured

The AS/400 system uses several electronic components to manage the transfer of data from a disk to main storage. Data must be in main storage before it can be used by a program.

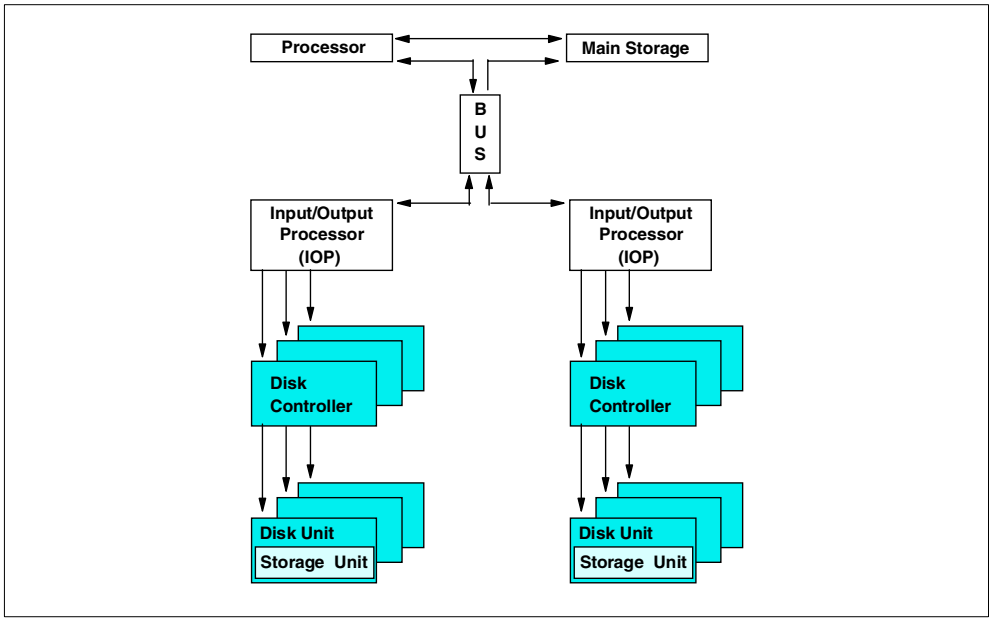


Figure 38. Components used for data transfer

Figure 38 shows the components that are used for data transfer. The components include:

- **Bus**: The bus is the main communications channel for input and output data transfer. A system may have one or more buses.
- **I/O processor**: The input/output processor (IOP) is attached to the bus. The IOP is used to transfer information between main storage and specific groups of controllers. Some IOPs are dedicated to specific types of controllers, such as disk controllers. Other IOPs can attach more than one type of controller (for example, tape controllers, and disk controllers).
- **Disk controller**: The disk controller attaches to the IOP and handles the information transfer between the IOP and the disk units. Some disk units have built-in controllers. Others have separate controllers.
- **Disk unit**: Disk units are the actual devices that contain the storage units. Hardware is ordered at the disk-unit level and each disk unit has a unique serial number.

A.2 Full protection: Single ASP

A simple and safe way to manage and protect your auxiliary storage is to perform the following tasks:

- Assign all disk units to a single auxiliary storage pool (the system ASP).
- Use device parity protection for all disk units that have the hardware capability.
- Use mirrored protection for the remaining disk units on the system.

With this method, your system continues to run even if a single disk unit fails. When the disk is replaced, the system can reconstruct the information so that no data is lost. The system may also continue to run when a disk-related hardware component fails. Whether your system continues to run depends on your

configuration. For example, the system continues to run if an IOP fails and all of the attached disk units have mirrored pairs that are attached to a different IOP.

When you use a combination of mirrored protection and device parity protection to fully protect your system, you increase your disk capacity requirements. Device parity protection requires up to 25% of the space on your disk units to store parity information. Mirrored protection doubles the disk requirement for all disks that do not have the device parity protection capability.

Figure 39 shows an example of a system with full protection. The system has 21 disk units. All of the disk units are assigned to the system ASP. The system assigns unit numbers to each configured disk on the system. Notice that the mirrored pairs share a common unit number.

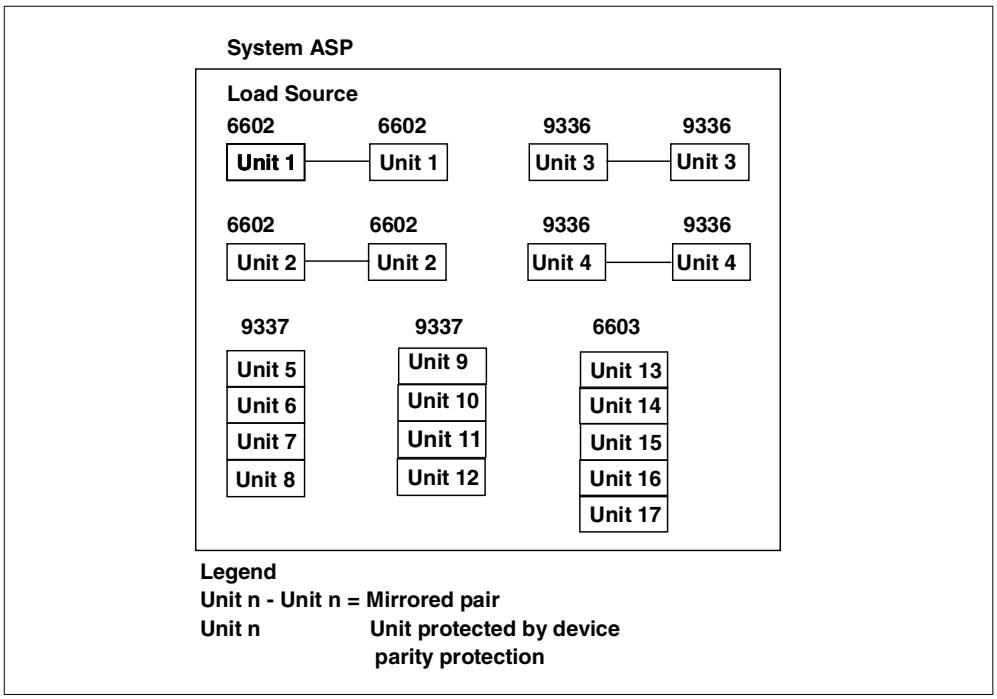


Figure 39. Full protection: Single ASP

A.3 Full protection: Multiple ASPs

You may want to divide your disk units into several auxiliary storage pools. Sometimes, your overall system performance may improve by having user ASPs.

For example, you can isolate journal receivers in a user ASP. Or, you can place history files or documents that seldom change in a user ASP that has lower performance disk units.

You can fully protect a system with multiple ASPs by performing the following tasks:

- Use device parity protection for all disk units that have the hardware capability.
- Set up mirrored protection for every ASP on the system. You can set up mirrored protection even for an ASP that has only disk units with device parity

protection. That way, if you add units that do not have device parity protection in the future, those units are automatically mirrored.

Note: You must add new units in pairs of units with equal capacity. Before configuring this level of protection, be sure that you know how to assign disk units to ASPs.

Figure 40 shows an example of two ASPs. Both ASPs have device parity protection and mirrored protection defined. Currently, ASP 2 has no mirrored units.

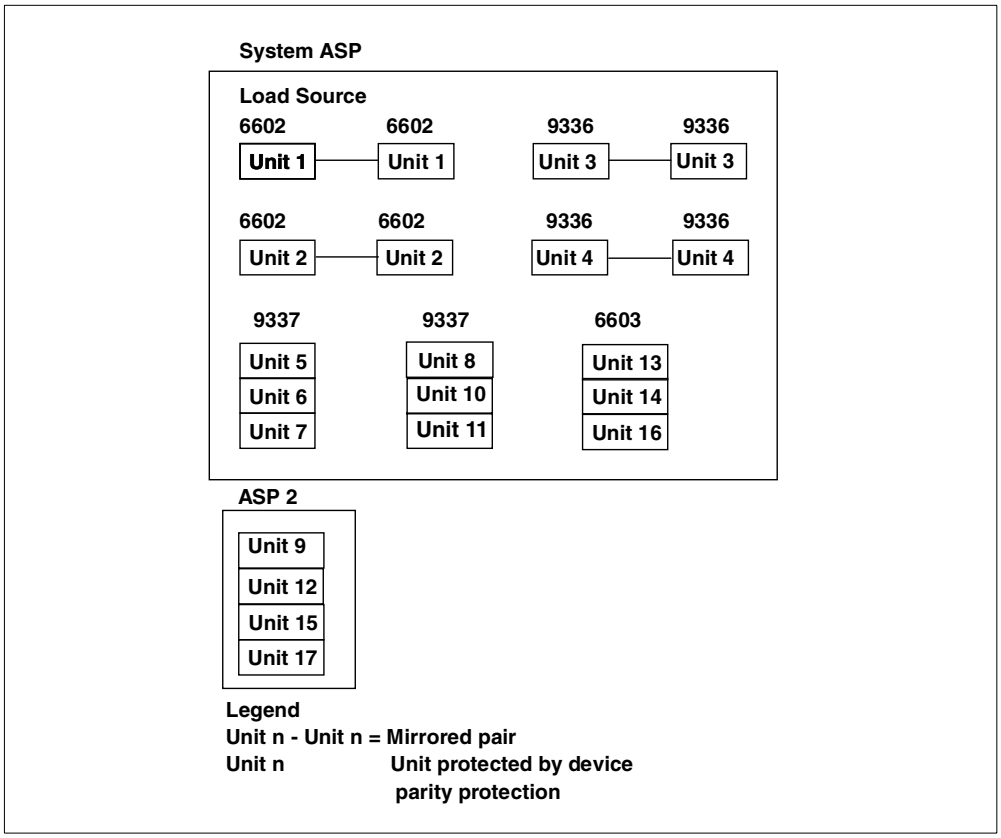


Figure 40. Full protection: Multiple ASPs

A.4 Partial protection: Multiple ASPs

Sometimes, full protection (using a combination of device parity protection and mirrored protection) may be too costly. If this happens, you need to develop a strategy to protect the critical information on your system. Your objectives should be to minimize the loss of data and to reduce the amount of time in which critical applications are not available. Your strategy should involve dividing your system into user ASPs and protecting only certain ASPs. Note, however, that if the system is not fully protected and an unprotected disk unit fails, serious problems can occur. The entire system can become unusable, end abnormally, require a long recovery, and data in the ASP that contains the failed unit must be restored.

Before configuring this level of protection, be sure that you know how to assign disk units to ASPs.

The following list provides suggestions for developing your strategy:

- If you protect the system ASP with a combination of mirrored protection and device parity protection, you can reduce or eliminate recovery time. The system ASP, and particularly the load source unit, contain information that is critical to keeping your system operational. For example, the system ASP has security information, configuration information, and addresses for all the libraries on the system.
- Think about how you can recover file information. If you have online applications, and your files change constantly, consider using journaling and placing journal receivers in a protected user ASP.
- Think about what information does not need protection. This is usually information that changes infrequently. For example, history files may need to be online for reference, but the data in the history files may not change except at the end of the month. You could place those files in a separate user ASP that does not have any disk protection. If a failure occurs, the system becomes unusable, but the files can be restored without any loss of data. The same may be true for documents.
- Think about other information that may not need disk protection. For example, your application programs may be in a separate library from the application data. It is likely the case that the programs change infrequently. The program libraries could be placed in a user ASP that is not protected. If a failure occurs, the system becomes unusable, but the programs can be restored.

Two simple guidelines can summarize the previous list:

1. To reduce recovery time, protect the system ASP.
2. To reduce loss of data, make conscious decisions about which libraries must be protected.

Figure 41 on page 146 shows an example of three ASPs. ASP 1 (system ASP) and ASP 3 have device parity protection and mirrored protection defined. Currently, ASP 3 has no mirrored units and ASP 2 has no disk protection. In this example, ASP 2 could be used for history files, reference documents, or program libraries. ASP 3 could be used for journal receivers and save files.

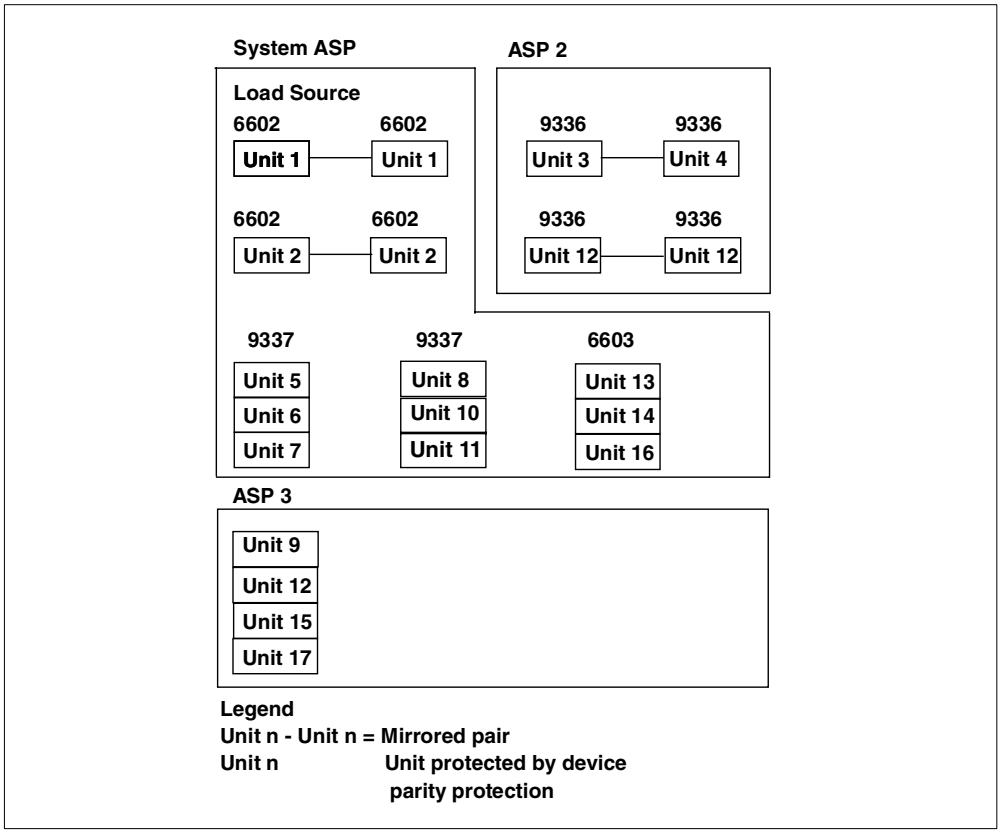


Figure 41. Different protection for multiple ASPs

Appendix B. Planning for device parity protection

If you intend to have a system with data loss protection and concurrent maintenance repair, plan to use one of the following configurations:

- Mirrored protection and device parity protection to protect the system ASP.
- Mirrored protection for the system ASP and device parity protection for user ASPs.
- Mirrored protection and device parity protection to protect the system ASP and user ASPs.

Note: You can use device parity protection with disk array subsystems as well as with input-output processors (IOP).

For each device parity protection set, the space that is used for parity information is equivalent to one disk unit. The minimum number of disk units in a subsystem with device parity protection is four. The maximum number of disk units in a subsystem with device parity protection is seven, eight, or 16, depending on the type. A subsystem with 16 disk units attached has two device parity protection sets and the equivalent of two disk units dedicated to parity information.

For more information about device parity protection, see *Backup and Recovery,* SC41-5304.

B.1 Mirrored protection and device parity protection to protect the system ASP

This section illustrates an example of a system with a single auxiliary storage pool (ASP). The ASP has both mirrored protection and device parity protection. When one of the disk units with device parity protection fails, the system continues to run. The failed unit can be repaired concurrently. If one of the mirrored disk units fails, the system continues to run using the operational unit of the mirrored pair. Figure 42 on page 148 shows an example of mirrored protection and device parity protection used in the system ASP.

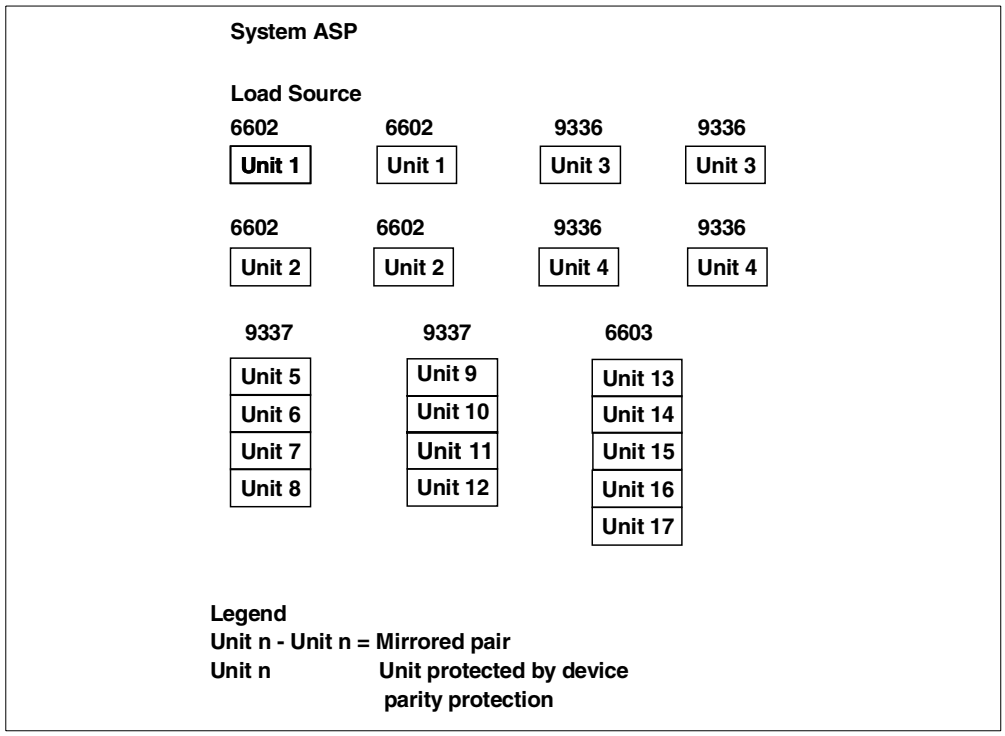


Figure 42. Mirrored protection and device parity protection to protect the system ASP

B.2 Mirrored protection in the system ASP and device parity protection in the user ASPs

You should consider device parity protection if you have mirrored protection in the system ASP and are going to create user ASPs. The system can tolerate a failure in one of the disk units in a user ASP. The failure can be repaired while the system continues to run. Figure 43 shows an example of a system ASP with device parity.

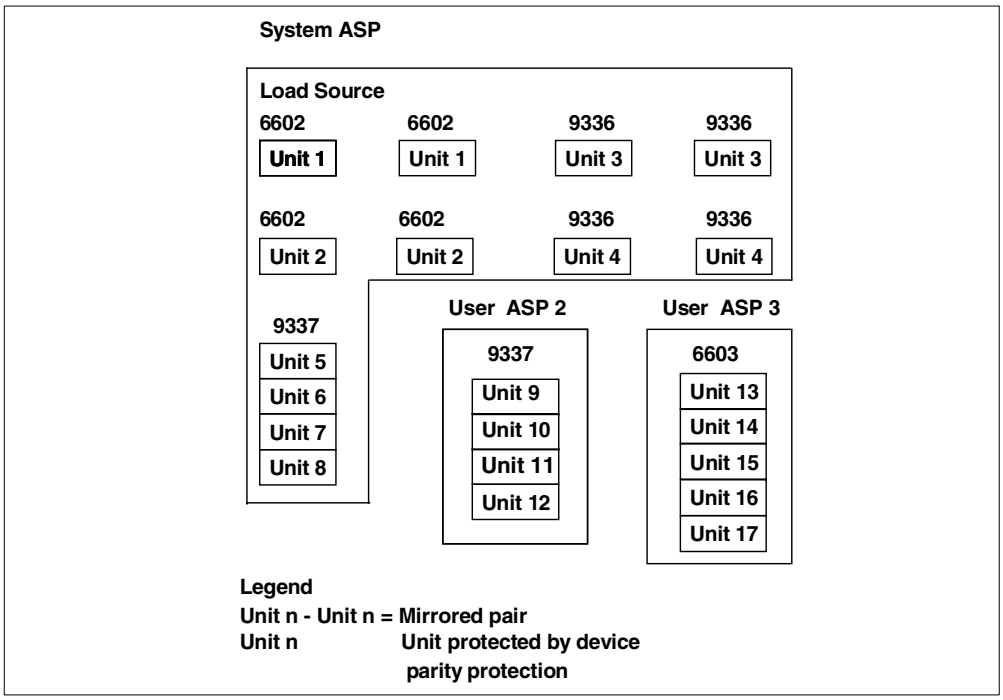


Figure 43. Mirrored protection in the system ASP and device parity protection in the user ASPs

B.2.1 Mirrored protection and device parity protection in all ASPs

If you have all ASPs protected with mirrored protection, to add units to the existing ASPs, also consider using device parity protection. The system can tolerate a failure in one of the disk units with device parity protection. The failed unit can be repaired while the system continues to run.

If a failure occurs on a disk unit that has mirrored protection, the system continues to run using the operational unit of the mirrored pair. Figure 44 on page 150 shows an example of mirrored protection and device parity protection in all ASPs.

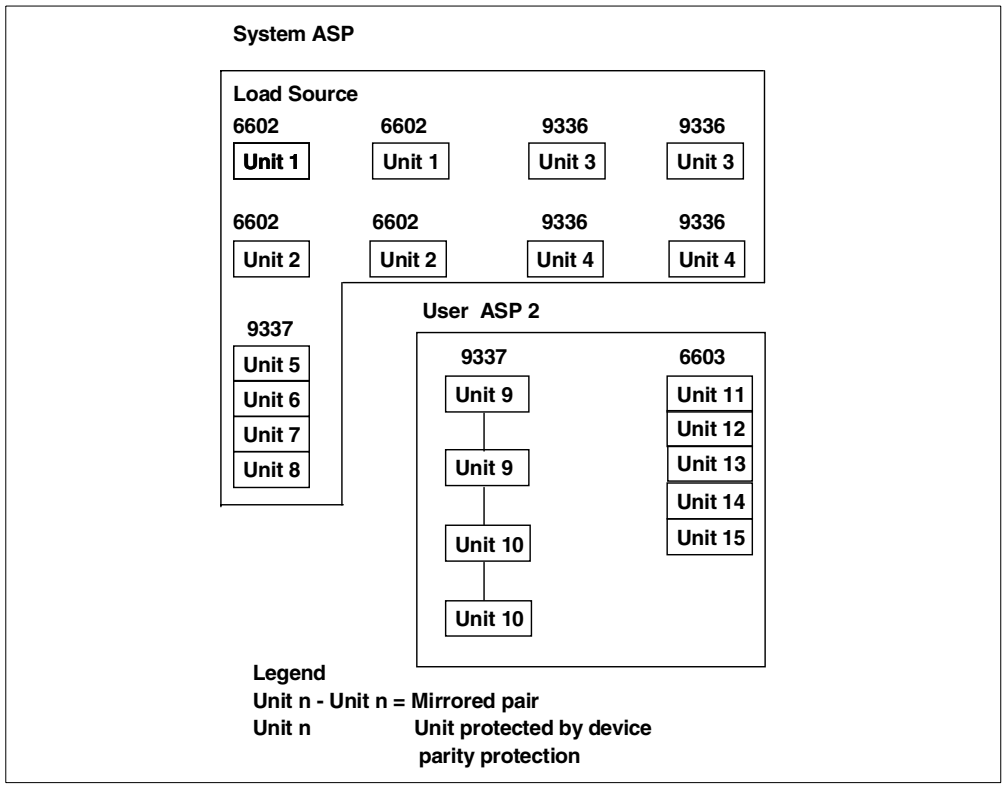


Figure 44. Mirrored protection and device parity protection in all ASPs

B.2.2 Disk controller and the write-assist device

The disk controller for the subsystems with device parity protection performs an important function for write operations. The controller keeps a list of all uncommitted data written to the write-assist device that has not been written to the data disk or the parity disk. Use this list during a power failure on the AS/400 system.

Write requests and the write-assist device

A write request to the subsystems with device parity protection starts three write operations. Data to be written to the disk units is first stored in the buffer in the disk controller. From this buffer, the data is sent to the write-assist device, the data disk, and the parity disk.

The following actions occur during a write request:

1. A write operation to the write-assist device:

   Data is written to the write-assist device sequentially. A write operation to the write-assist device does not require parity calculation.

   The disk controller (identifier and disk address) adds the header information. Trailing information is added for the data before writing to the write-assist device. The header information can be used during a power failure.

   Normally, the write operations to the write-assist device are completed before the write operations to the disk units. The disk controller sends a completion message to storage management that allows the application to continue. The

data that is written on the write-assist device is marked as uncommitted on the disk controller.

Note: The write operation to the data disk and the parity disk continues in the background until the data is successfully written and is marked as *committed* in the disk controller.

2. A write operation to the disk unit.
   - For data, the operation:
     – Reads the original data
     – Writes the new data
   - For parity data, the operation:
     – Reads the original parity information
     – Compares the new data with the original data and the original parity to calculate the new parity
     – Writes the new parity information

   The write operation to the data disk usually completes before the write operation to the parity disk. The write operation to the data disk does not have to wait for the parity calculation. The delay between the writing of new data and the writing of the new parity information is known as *delayed parity*.
3. Data is marked as committed data when it is successfully written to both the data disk unit and the parity disk unit.
4. A completion message is sent to storage management only if the write operation on the write-assist device or the data disk unit has not already sent a message.

The performance for this type of write operation depends on disk contention and the time that is needed to calculate the parity information.

B.2.3 Mirrored protection: How it works

Since mirrored protection is configured by ASP, all ASPs must be mirrored to provide for maximum system availability. If a disk unit fails in an ASP that is not mirrored, the system can't be used until the disk unit is repaired or replaced.

The start mirrored pairing algorithm automatically selects a mirrored configuration to provide the maximum protection at the bus, I/O processor, or controller level for the hardware configuration of the system. When storage units of a mirrored pair are on separate buses, they have maximum independence or protection. Because they do not share any resource at the bus, I/O processor, or controller levels, a failure in one of these hardware components allows the other mirrored unit to continue operating.

Any data written to a unit that is mirrored is written to both storage units of the mirrored pair. When data is read from a unit that is mirrored, the read operation can be from either storage unit of the mirrored pair. Because it is transparent to the user, they don't know from which mirrored unit the data is being read. The user is also not aware of the existence of two physical copies of the data.

If one storage unit of a mirrored pair fails, the system suspends mirrored protection to the failed mirrored unit. The system continues to operate using the

remaining mirrored unit. The failing mirrored unit can be physically repaired or replaced.

After the failed mirrored unit is repaired or replaced, the system synchronizes the mirrored pair by copying current data from the storage unit that has remained operational to the other storage unit. During synchronization, the mirrored unit to which the information is being copied is in the resuming state. Synchronization does not require a dedicated system and runs concurrently with other jobs on the system. System performance is affected during synchronization. When synchronization is complete, the mirrored unit becomes active.

Appendix C. Batch Journal Caching for AS/400 boosts performance

In the year 2000, a Programming Request for Price Quotation (PRPQ) offering became available to improve the performance of an AS/400e system when journals are involved. It is known as Batch Journal Caching for AS/400 PRPQ, and the order number is 5799-BJC. It installs and runs correctly on any national language version.

C.1 Overview

The Batch Journal Caching for AS/400 PRPQ can provide a significant performance improvement for batch environments that use journaling. Benefits include:

- It changes the handling of disk writes to achieve the maximum performance for journaled database operations.
- By caching journal writes in main memory, it can greatly reduce the impact of journaling on batch run time by eliminating the delay in waiting for each journal entry to be written to disk.

This PRPQ is an ideal solution for customers with batch workloads who use journaling as part of a high availability solution to replicate database changes to a backup system.

C.2 Benefits of the Batch Journal Caching PRPQ

Applications that perform large numbers of database add, update, or delete operations should experience the greatest improvement when this PRPQ is active. Although it is directed primarily toward batch jobs, some interactive applications may also benefit. Applications using commitment control should see less improvement because commitment control already performs some journal caching.

With traditional non-cached journaling in a batch environment, each database record added, updated, or deleted by the batch job causes a new journal entry to be constructed in main memory. The batch job then waits for each new journal entry to be written to disk to assure recovery. This results in a large number of disk writes.

The Batch Journal Caching PRPQ provides the ability to selectively enable a new variation of journal caching. It changes the handling of disk writes to achieve the maximum performance for journaled database operations. Both the journal entries and the corresponding database records are cached in main memory, thereby delaying writing journal entries to disk until an efficient disk write can be scheduled. This prevents most database operations from being held up while waiting for the synchronous write of journal entries to disk.

By more aggressively caching journal writes in main memory, it can:

- Greatly reduce the impact of journaling on batch run time by reducing the delay in waiting for each journal entry to be written to disk.

- Avoid the problems and costs associated with making application changes (such as adding commitment control) to improve batch performance in these environments.

C.2.1 Optimal journal performance

For optimal journal performance, many factors beyond using this PRPQ should be considered, including:

- The number and type of disk units and disk controllers
- Amount of write cache
- Placement of journal receivers in user auxiliary storage pools (ASPs)
- Application changes

C.3 Installation considerations

The prerequisites and limitations of the Batch Journal Cache PRPQ are identified here.

C.3.1 Prerequisites

This PRPQ runs under the latest levels of operating system. Install either:

- OS/400 V4R5 with PTFs MF24863, MF24866, MF24870, and SF63192
- OS/400 V4R4 with PTFs MF24293, MF24626, and SF63186

C.3.2 Limitations

This batch cache type of journaling differs from traditional journaling and can affect the recover ability of the associated database files. Because journal entries are temporarily cached in main memory, a few recent database changes that are still cached and not yet written to disk can be lost in the rare event of a severe system failure where the contents of main memory are not preserved.

This type of journaling may not be suitable for interactive applications where single system recovery is the primary reason for using journaling. Also, it may not be suitable where it is unacceptable to lose even one recent database change in the rare event of a system failure in which the contents of main memory are not preserved.

Batch journal caching is primarily intended for situations where journaling is used to enable database replication to a second system (for example, for high availability or Business Intelligence applications) under a heavy workload like batch processing. This also applies to heavy interactive work with applications that do not employ commitment control. This function can also be selectively enabled to optimize performance when running nightly batch workloads. It can then disabled each morning to optimize recoverability when running interactive applications. This can speed up some nightly batch jobs without sacrificing robust recovery for daytime interactive work.

C.3.3 For more information

After installing the PRPQ software product, read the README member of the README AS/400 file in the QBJC library for instructions on this product. Use DSPPFM FILE(QBJC/README) MBR(README) to display the file.

For further information, contact your IBM marketing representative.

Appendix D. Sample program to calculate journal size requirement

D.1 ESTJRNSIZ CL program

```
esj1: pgm
      dclf estjrnsiz/lastipl
      call qwccrtec /* retrieve last IPL information */
      CPYSPLF FILE(QPSRVDMP) TOFILE(ESTJRNSIZ/LASTIPL)
      DLTSPLF FILE(QPSRVDMP)
 loop: rcvf
       monmsg msgid(CPF0864) exec(goto cmdlbl(endit))
       if (%sst(&lastipl 103 17) *ne '                 ') +
          then(chgdtaara lastipl %sst(&lastipl 103 17))
       goto loop
 endit: call pfildtl
      endpgm
```

D.2 NJPFILS RPGLE program

```
    FQPRINT    O    F  132        PRINTER OFLIND(*INOF) USROPN
    FPFILRPT   O    E             Printer OFLIND(*IN90)
    D* Include error code parameter
    D/COPY QSYSINC/QRPGLESRC,QUSEC
    Dlstlib           s             10A
    Dlstfil           s             10A
    Dipltim           s               z
    Dtimipl           ds
    D ccipl                          2A
    D yyipl                          2A
    D sep1                           1A   INZ('-')
    D mmipl                          2A
    D sep2                           1A   INZ('-')
    D ddipl                          2A
    D sep3                           1A   INZ('-')
    D hhipl                          2A
    D sep4                           1A   INZ('.')
    D nnipl                          2A
    D sep5                           1A   INZ('.')
    D ssipl                          2A
    D sep6                           1A   INZ('.')
    D msipl                          6A   INZ('000000')
    Dlastipl          ds
    D iplmo                   1      2A
    D iplda                   4      5A
    D iplyr                   7      8A
    D iplhr                  10     11A
    D iplmi                  13     14A
    D iplse                  16     17A
    Dtimestamp        s               z   INZ(*SYS)
    Dqmbrovr          s              1A
    Dqmbrfmt          s              8A
    Dqmbrdovr         s              9B 0 INZ(4096)
    Dnummbrs          s              4B 0
    Dnumobjs          s              4B 0
    Dobjtolist        s             20    INZ('*ALL      *ALLUSR   ')
    DFIRST_ERR        S              1    INZ('0')
    Dobj_count        s              9  0 INZ(1)
    Dmbr_count        s              9  0 INZ(1)
    Dobjspcnam        s             20A   INZ('OBJECTS   ESTJRNSIZ ')
    Dmbrspcnam        s             20A   INZ('MEMBERS   ESTJRNSIZ ')
    Dext_attr         s             10A
    Dspc_name         s             20A
    Dspc_size         s              9B 0 INZ(1)
    Dspc_init         s              1    INZ(x'00')
    Dobjlstptr        s               *
    Dmbrlstptr        s               *
    Dobjspcptr        s               *
    Dmbrspcptr        s               *
    DARR              s              1    BASED(objlstptr) DIM(32767)
    DRCVVAR           s              8
    DRCVVARSIZ        s              9B 0 INZ(8)
    DARRm             s              1    BASED(mbrlstptr) DIM(32767)
    DRCVVARm          s              8
```

```
DRCVVARSIZm       s              9B 0 INZ(8)
D* Common list header
DOUSH0100         DS                  BASED(OBJSPCPTR)
D OUSUA                    1     64                                    User area
D OUSSGH                  65     68B 0                                 Size generic header
D OUSSRL                  69     72                                    Structure rel level
D OUSFN                   73     80                                    Format name
D OUSAU                   81     90                                    API used
D OUSDTC                  91    103                                    Date time created
D OUSIS                  104    104                                    Information status
D OUSSUS                 105    108B 0                                 Size user space
D OUSOIP                 109    112B 0                                 Offset input parm
D OUSSIP                 113    116B 0                                 Size input parm
D OUSOHS                 117    120B 0                                 Offset header secti
D OUSSHS                 121    124B 0                                 Size header section
D OUSOLD                 125    128B 0                                 Offset list data
D OUSSLD                 129    132B 0                                 Size list data
D OUSNBRLE               133    136B 0                                 Number list entries
D OUSSEE                 137    140B 0                                 Size each entry
D OUSSIDLE               141    144B 0                                 CCSID list ent
D QUSCID                 145    146                                    Country ID
D OUSLID                 147    149                                    Language ID
D OUSSLI                 150    150                                    Subset list indicat
D OUSERVED00             151    192                                    Reserved
D* Common list header
DMUSH0100         DS                  BASED(MBRSPCPTR)
D MUSUA                    1     64                                    User area
D MUSSGH                  65     68B 0                                 Size generic header
D MUSSRL                  69     72                                    Structure rel level
D MUSFN                   73     80                                    Format name
D MUSAU                   81     90                                    API used
D MUSDTC                  91    103                                    Date time created
D MUSIS                  104    104                                    Information status
D MUSSUS                 105    108B 0                                 Size user space
D MUSOIP                 109    112B 0                                 Offset input parm
D MUSSIP                 113    116B 0                                 Size input parm
D MUSOHS                 117    120B 0                                 Offset header secti
D MUSSHS                 121    124B 0                                 Size header section
D MUSOLD                 125    128B 0                                 Offset list data
D MUSSLD                 129    132B 0                                 Size list data
D MUSNBRLE               133    136B 0                                 Number list entries
D MUSSEE                 137    140B 0                                 Size each entry
D MUSSIDLE               141    144B 0                                 CCSID list ent
D MUSCID                 145    146                                    Country ID
D MUSLID                 147    149                                    Language ID
D MUSSLI                 150    150                                    Subset list indicat
D MUSERVED00             151    192                                    Reserved
D* Structure for OBJL0200
DQUSL020002       DS                  BASED(objlstptr)
D QUSOBJNU00               1     10                                    Object name used
D QUSOLNU00               11     20                                    Object lib name use
D QUSOBJTU00              21     30                                    Object type used
D QUSIS01                 31     31                                    Information status
D QUSEOA                  32     41                                    Extended object attr
D QUSTD06                 42     91                                    Text description
D QUSUDA                  92    101                                    User defined attr
D QUSERVED22             102    108                                    Reserved
D* File Definition Template (FDT) Header
D*   This section is always located at the beginning of the returned data.
DQDBQ25           DS          4096                                    Header info
D QDBFYRET                 1      4B 0                                 Bytes returned
D QDBFYAVL                 5      8B 0                                 Bytes available
D*QDBFHFLG                 2
D  QDBBITS27               9     10                                    Attribute bytes
D  QDBBITS1                9      9
D*  QDBRSV100              2           BITS
D*  QDBFHFPL00             1           BIT
D*  QDBRSV200              1           BIT
D*  QDBFHFSU00             1           BIT
D*  QDBRSV300              1           BIT
D*  QDBFHFKY00             1           BIT
D*  QDBRSV400              1           BIT
D*  QDBFHFLC00             1           BIT
D*  QDBFKFSO00             1           BIT
D*  QDBRSV500              1           BIT
D*  QDBFHSHR00             1           BIT
D*  QDBRSV600              2           BITS
D*  QDBFIGCD00             1           BIT
```

```
D*  QDBFIGCL00           1          BIT
D QDBRSV7               11     14                                      reserved
D QDBLBNUM              15     16B 0                                   # data members
D*QDBFKDAT                     14
D  QDBFKNUM00           17     18B 0
D  QDBFKMXL00           19     20B 0
D* QDBFKFLG00                   1
D   QDBBITS28           21     21
D*  QDBRSV802            1          BIT
D*  QDBFKFCS02           1          BIT
D*  QDBRSV902            4          BITS
D*  QDBFKFRC02           1          BIT
D*  QDBFKFLT02           1          BIT
D  QDBFKFDM00           22     22
D  QDBRSV1000           23     30                                      keyed seq ap
D QDBFHAUT              31     40                                      public aut
D QDBFHUPL              41     41                                      pref storage unit
D QDBFHMXM              42     43B 0                                   max members
D*                               Maximum Members (MAXMBRS)
D QDBFWTFI              44     45B 0                                   max file wait time
D QDBFHFRT              46     47B 0                                   FRCRATION
D QDBHMNUM              48     49B 0                                   # members
D QDBRSV11              50     58                                      reserved
D*                               Reserved.
D QDBFBRWT              59     60B 0                                   max recd wait time
D*QDBQAAF00                    1
D  QDBBITS29            61     61                                      add'l attrib flags
D*  QDBRSV1200          7          BITS
D*  QDBFPGMD00           1          BIT
D QDBMTNUM              62     63B 0                                   tot # recd fmts
D*QDBFHFL2                     2
D  QDBBITS30            64     65                                      add'l attrib flags
D*  QDBFJNAP00           1          BIT
D*  QDBRSV1300           1          BIT
D*  QDBFRDCP00           1          BIT
D*  QDBFWTCP00           1          BIT
D*  QDBFUPCP00           1          BIT
D*  QDBFDLCP00           1          BIT
D*  QDBRSV1400           9          BITS
D*  QDBFKFND00           1          BIT
D QDBFVRM               66     67B 0                                   1st supported VRM
D  QDBBITS31            68     69                                      add'l attrib flags
D*  QDBFHMCS00           1          BIT
D*  QDBRSV1500           1          BIT
D*  QDBFKNLL00           1          BIT
D*  QDBFNFLD00           1          BIT
D*  QDBFVFLD00           1          BIT
D*  QDBFTFLD00           1          BIT
D*  QDBFGRPH00           1          BIT
D*  QDBFPKEY00           1          BIT
D*  QDBFUNQC00           1          BIT
D*  QDBR11800            2          BITS
D*  QDBFAPSZ00           1          BIT
D*  QDBFDISF00           1          BIT
D*  QDBR11900            3          BITS
D QDBFHCRT              70     82                                      file level indicato
D  QDBRSV1800           83     84
D  QDBFHTXT00           85    134                                      file text descript
D QDBRSV19             135    147                                      reserved
D*QDBFSRC                      30
D  QDBFSRCF00          148    157
D  QDBFSRCM00          158    167
D  QDBFSRCL00          168    177                                      source file fields
D*                               Source File Fields
D QDBFKRCV             178    178                                      access path recover
D QDBRSV20             179    201                                      reserved
D QDBFTCID             202    203B 0                                   CCSID
D QDBFASP              204    205                                      ASP
D*                               X'0000' = The file is located in the system ASP
D*                               X'0002'-X'0010' = The user ASP the file is located in.
D  QDBBITS71           206    206                                      complex obj flags
D*  QDBFHUDT00           1          BIT
D*  QDBFHLOB00           1          BIT
D*  QDBFHDTL00           1          BIT
D*  QDBFHUDF00           1          BIT
D*  QDBFHLON00           1          BIT
D*  QDBFHLOP00           1          BIT
D*  QDBFHDLL00           1          BIT
```

```
D*  QDBRSV2101            1          BIT
D QDBXFNUM              207    208B 0                                  max # fields
D QDBRSV22              209    284                                     reserved
D QDBFODIC              285    288B 0                                  offset to IDDU/SQL
D QDBRSV23              289    302                                     reserved
D QDBFFIGL              303    304B 0                                  file generic key
D QDBFMXRL              305    306I 0                                  max record len
D FMXRL1                305    305A
D QDBRSV24              307    314                                     reserved
D QDBFGKCT              315    316B 0                                  file generic key
D                                                                       field count
D QDBFOS                317    320B 0                                  offset to file scop
D                                                                       array
D QDBRSV25              321    328                                     reserved
D QDBFOCS               329    332B 0                                  offset to alternate
D                                                                       collating sequence
D                                                                       table
D QDBRSV26              333    336                                     reserved
D QDBFPACT              337    338                                     access path type
D QDBFHRLS              339    344                                     file version/releas
D QDBRSV27              345    364                                     reserved
D QDBPFOF               365    368B 0                                  offset to pf speci-
D                                                                       fic attrib section
D QDBLFOF               369    372B 0                                  offset to LF speci-
D                                                                       fic attrib section
D   QDBBITS58           373    373
D*  QDBFSSCS02            3          BITS
D*  QDBR10302             5          BITS
D  QDBFLANG01           374    376
D  QDBFCNTY01           377    378                                     sort sequence table
D QDBFJORN              379    382B 0                                  offset to jrn
D                                                                       section
D*                              Journal Section, Qdbfjoal.
D QDBFEVID              383    386B 0                                  initial # distinct
D                                                                       values an encoded
D                                                                       vector AP allowed
D QDBRSV28              387    400                                     reserved
D*The FDT header ends here.
D*Journal Section
D*This section can be located with the offset Qdbfjorn, which is located in the FDT heade
DQDBQ40           DS                                                    jrn section
D QDBFOJRN               1     10                                      jrn nam
D QDBFOLIB              11     20                                      jrn lib nam
D*QDBFOJPT                      1
D  QDBBITS41            21     21                                      jrn options flags
D*  QDBR10600            1          BIT
D*  QDBFJBIM00           1          BIT
D*  QDBFJAIM00           1          BIT
D*  QDBR10700            1          BIT
D*  QDBFJOMT00           1          BIT
D*  QDBR10800            3          BITS
D QDBFJACT              22     22                                      jrn options
D*                               '0' = The file is not being journaled
D*                               '1' = The file is being journaled
D QDBFLJRN              23     35                                      last jrn-ing date
D QDBR105               36     64                                      reserved
D* Structures for QDBRTVFD
D* Input structure for QDBRTVFD API header section
DQDBRIP           DS                                                    Qdb Rfd Input Parms
D*QDBRV                  1      1                                       varying length
D QDBLORV                2      5B 0                                    Len. o rcvr var
D QDBRFAL                6     25                                       Ret'd file & lib
D QDBFN00               26     33                                       Format name
D QDBFALN               34     53                                       File & lib name
D QDBRFN00              54     63                                       Recd fmt name
D QDBFILOF              64     64                                       File override flag
D QDBYSTEM              65     74                                       System
D QDBFT                 75     84                                       Format type
D*QDBEC                 85     85                                       varying length
D* Retrieve member information structure
D*Type Definition for the MBRL0100 format of the userspace in the QUSLMBR API
DQUSL010000       DS                  BASED(mbrlstptr)
D QUSMN00                1     10                                      Member name
D*Record structure for QUSRMBRD MBRD0200 format
DQUSM0200         DS          4096
D QUSBRTN03              1      4B 0                                    Bytes Returned
D QUSBAVL04              5      8B 0                                    Bytes Available
D QUSDFILN00             9     18                                       Db File Name
```

```
D QUSDFILL00             19     28                                      Db File Lib
D QUSMN03                29     38                                      Member Name
D QUSFILA01              39     48                                      File Attr
D QUSST01                49     58                                      Src Type
D QUSCD03                59     71                                      Crt Date
D QUSSCD                 72     84                                      Src Change Date
D QUSTD04                85    134                                      Text Desc
D QUSSFIL01             135    135                                      Src File
D QUSEFIL               136    136                                      Ext File
D QUSLFIL               137    137                                      Log File
D QUSOS                 138    138                                      Odp Share
D QUSERVED12            139    140                                      Reserved
D QUSNBRCR              141    144B 0                                   Num Cur Rec
D QUSNBRDR              145    148B 0                                   Num Dlt Rec
D QUSDSS                149    152B 0                                   Dat Spc Size
D QUSAPS                153    156B 0                                   Acc Pth Size
D QUSNBRDM              157    160B 0                                   Num Dat Mbr
D QUSCD04               161    173                                      Change Date
D QUSSD                 174    186                                      Save Date
D QUSRD                 187    199                                      Rest Date
D QUSED                 200    212                                      Exp Date
D QUSNDU                213    216B 0                                   Nbr Days Used
D QUSDLU                217    223                                      Date Lst Used
D QUSURD                224    230                                      Use Reset Date
D QUSRSV101             231    232                                      Reserved1
D QUSDSSM               233    236B 0                                   Data Spc Sz Mlt
D QUSAPSM               237    240B 0                                   Acc Pth Sz Mlt
D QUSMTC                241    244B 0                                   Member Text Ccsid
D QUSOAI                245    248B 0                                   Offset Add Info
D QUSLAI                249    252B 0                                   Length Add Info
D QUSNCRU               253    256U 0                                   Num Cur Rec U
D QUSNDRU               257    260U 0                                   Num Dlt Rec U
D QUSRSV203             261    266                                      Reserved2
D* Record structure for data space activity statistics
DQUSQD            DS
D QUSNBRAO               1      8I 0                                   Num Act Ops
D QUSNBRDO               9     16I 0                                   Num Deact Ops
D QUSNBRIO              17     24I 0                                   Num Ins Ops
D QUSNBRUO              25     32I 0                                   Num Upd Ops
D QUSNBRDO00            33     40I 0                                   Num Del Ops
D QUSNBRRO00            41     48I 0                                   Num Reset Ops
D QUSNBRCO              49     56I 0                                   Num Cpy Ops
D QUSNBRRO01            57     64I 0                                   Num Reorg Ops
D QUSNAPBO              65     72I 0                                   Num APBld Ops
D QUSNBRLO              73     80I 0                                   Num Lrd Ops
D QUSNBRPO              81     88I 0                                   Num Prd Ops
D QUSNBRRK              89     96I 0                                   Num Rej Ksel
D QUSNRNK               97    104I 0                                   Num Rej NKsel
D QUSNRGB              105    112I 0                                   Num Rej Grp By
D QUSNBRIV             113    116U 0                                   Num Index Val
D QUSNBRII             117    120U 0                                   Num Index Ival
D QUSVDS               121    124U 0                                   Var Data Size
D QUSRSV107            125    192                                      Reserved 1
C* Set things up
C                   EXSR      INIT
C* Start mainline process
C*     set pointer to first object
C                   EVAL      objlstptr = objspcptr                     pt to b1 of usrsp
C                   EVAL      objlstptr = %addr(arr(OUSOLD + 1))        pt to entry 1
C                   EVAL      numobjs = OUSNBRLE
C*     process all entries
C                   DO        numobjs
C                   EVAL      libn = qusolnu00
C                   EVAL      filn = QUSOBJNU00
C                   IF        QUSEOA = 'PF        '                     only PF types
C                   EVAL      QDBFALN = filn + libn
C                   CALL      'QDBRTVFD'                                get full details
C                   parm                QDBQ25
C                   parm      4096      QDBLORV
C                   PARM                QDBRFAL
C                   parm      'FILD0100'  QDBFN00
C                   parm                QDBFALN
C                   parm      '*FIRST   '  QDBRFN00
C                   parm      '0'       QDBFILOF
C                   parm      '*LCL     '  QDBYSTEM
C                   parm      '*EXT     '  QDBFT
C                   parm                QUSEC
C                   IF        QUSBAVL > 0
```

```
C                   MOVEL     'QDBRTVFD'    APINAM           10
C                   EXSR      APIERR
C                   END
C* Have FD info, test for SRC versus DTA
C                   testb     '4'           QDBBITS1             10  11    10 on = DTA
C*                                                                         11 on = SRC
C     *in10         ifeq      *on                                         is a data file
C                   setoff                                       10
C* is the file already journaled?
C     QDBFJORN      ifgt      0
C* yes, get jrn info
C                   eval      QDBQ40 = %SUBST ( QDBQ25
C                                     : QDBFJORN + 1
C                                     : %SIZE( QDBQ40 ))
C                   eval      jrnnam = QDBFOJRN
C                   eval      jrnlib = QDBFOLIB
C                   if        QDBFJACT = '0'
C                   eval      jrnact = 'N'
C                   end
C                   if        QDBFJACT = '1'
C                   eval      jrnact = 'Y'
C                   end
C                   else
C                   eval      jrnnam = *blanks
C                   eval      jrnlib = *blanks
C                   eval      jrnact = ' '
C                   end
C* now get member list
C                   EVAL      spc_name = mbrspcnam
C                   CALL      'QUSLMBR'
C                   parm                spc_name
C                   parm      'MBRL0100'    mbr_fmt           8
C                   parm                QDBFALN
C                   parm      '*ALL    '    mbr_nam          10
C                   parm      '0'           mbr_ovr           1
C                   parm                QUSEC
C                   IF        QUSBAVL > 0                                 Any errors?
C                   MOVEL     'QUSLMBR'     APINAM
C                   EXSR      APIERR
C                   END
C* resolve pointer
C                   CALL      'QUSPTRUS'
C                   PARM                SPC_NAME
C                   PARM                MBRSPCPTR
C                   PARM                QUSEC
C* Check for errors on QUSPTRUS
C     QUSBAVL       IFGT      0
C                   MOVEL     'QUSPTRUS'    APINAM           10
C                   EXSR      APIERR
C                   END
C                   EVAL      mbrlstptr = mbrspcptr
C                   EVAL      mbrlstptr = %addr(arrm(MUSOLD + 1))
C                   EVAL      nummbrs = MUSNBRLE
C                   DO        nummbrs
C                   EVAL      mbrn = QUSMN00
C                   EVAL      QDBFALN = FILN + LIBN
C                   CALL      'QUSRMBRD'
C                   PARM                QUSM0200
C                   PARM                QMBRDOVR
C                   parm      'MBRD0200'    QMBRFMT
C                   parm                QDBFALN
C                   parm                QUSL010000
C                   parm      '0'           QMBROVR
C                   parm                QUSEC
C                   IF        QUSBAVL > 0
C                   MOVEL     'QUSRMBRD'    APINAM           10
C                   EXSR      APIERR
C                   END
C                   eval      QUSQD = %SUBST ( QUSM0200
C                                     : QUSOAI + 1
C                                     : QUSLAI )
C* have detail info, now create data
C                   eval      rcdlen = qdbfmxrl
C* calc seconds since IPL
C     timestamp     subdur    ipltim        runsec:*S        10 0
C* calc ave ops per sec
C     QUSNBRIO      add       QUSNBRUO      rcdops
C                   eval      rcdops = rcdops + QUSNBRDO
```

```
C     QUSNBRRO00    add       QUSNBRRO01    mbrops
C     rcdops        IFGT      0
C     mbrops        ORGT      0
C     rcdops        div       runsec        avrrcdops
C     mbrops        div       runsec        avrmbrops
C     avrrcdops     add       avrmbrops     jrnsec
C     rcdlen        add       155           jrnsiz
C                   eval      jrnsiz = (jrnsiz * jrnsec * 86400) / 1048576
C                   END
C                   EXSR      dodetail
C                   eval      avrrcdops = 0
C                   eval      avrmbrops = 0
C                   eval      jrnsec = 0
C                   eval      jrnsiz = 0
C                   EVAL      mbrlstptr = %addr(arrm(MUSSEE + 1))           incr to next ent
C                   END
C                   END
C                   END
C                   EVAL      objlstptr = %addr(arr(OUSSEE + 1))
C                   END
C* End mainline process
C                   EXSR      DONE
C* * * Subroutines follow * * *
C* INIT subroutine
C     INIT          BEGSR
C                   OPEN      QPRINT
C                   exsr      wrthead
C                   z-add     16            qusbprv                       set err code struct
C                                                                         to omit exceptions
C* Does user space exist for OBJECT list?
C                   eval      spc_name = objspcnam
C                   eval      ext_attr = 'QUSLOBJ   '
C                   EXSR      USRSPC
C* Does user space exist for MEMBER list?
C                   eval      spc_name = mbrspcnam
C                   eval      ext_attr = 'QUSLMBR   '
C                   EXSR      USRSPC
C* Retrieve last IPL time derived from previous step
C     *DTAARA       DEFINE    LASTIPL       LASTIPL          17
C                   IN        LASTIPL
C                   move      iplyr         yripl             2 0
C                   move      iplyr         yyipl
C                   move      iplmo         mmipl
C                   move      iplda         ddipl
C                   move      iplhr         hhipl
C                   move      iplmi         nnipl
C                   move      iplse         ssipl
C                   IF        yripl > 88
C                   move      '19'          ccipl
C                   ELSE
C                   move      '20'          ccipl
C                   END
C                   MOVEL     TIMIPL        IPLTIM
C* Fill the user space with object list
C                   eval      spc_name = objspcnam
C                   call      'QUSLOBJ'
C                   parm                    spc_name
C                   parm      'OBJL0200'    fmtnam            8
C                   parm                    objtolist
C                   parm      '*FILE     '  objtype          10
C                   parm                    QUSEC
C* Any errors?
C                   IF        QUSBAVL > 0
C                   MOVEL     'QUSLOBJ'     APINAM
C                   EXSR      APIERR
C                   END
C* Get a resolved pointer to the user space
C                   CALL      'QUSPTRUS'
C                   PARM                    SPC_NAME
C                   PARM                    OBJSPCPTR
C                   PARM                    QUSEC
C* Check for errors on QUSPTRUS
C     QUSBAVL       IFGT      0
C                   MOVEL     'QUSPTRUS'    APINAM           10
C                   EXSR      APIERR
C                   END
C                   ENDSR
C*
```

```
C* USRSPC subroutine
C     USRSPC        BEGSR
C* Verify user space exists
C                   CALL      'QUSROBJD'
C                   PARM                    RCVVAR
C                   PARM                    RCVVARSIZ
C                   PARM      'OBJD0100'    ROBJD_FMT         8
C                   PARM                    SPC_NAME
C                   PARM      '*USRSPC'     SPC_TYPE         10
C                   PARM                    QUSEC
C* Errors on QUSROBJD?
C                   IF        QUSBAVL > 0
C                   IF        QUSEI = 'CPF9801'                              user space not foun
C                   CALL      'QUSCRTUS'                                     create the space
C                   PARM                    SPC_NAME
C                   PARM                    EXT_ATTR         10
C                   PARM                    SPC_SIZE
C                   PARM                    SPC_INIT
C                   PARM      '*ALL'        SPC_AUT          10
C                   PARM      *BLANKS       SPC_TEXT         50
C                   PARM      '*YES'        SPC_REPLAC       10
C                   PARM                    QUSEC
C                   PARM      '*USER'       SPC_DOMAIN       10
C* Errors on QUSCRTUS?
C                   IF        QUSBAVL > 0
C                   MOVEL     'QUSCRTUS'    APINAM           10
C                   EXSR      APIERR
C                   END
C*   else error occurred accessing the user space
C                   ELSE
C                   MOVEL     'QUSROBJD'    APINAM           10
C                   EXSR      APIERR
C                   END
C                   END
C                   ENDSR
C* APIERR subroutine
C     APIERR        BEGSR
C* If first error found, then open QPRINT *PRTF
C                   IF        NOT %OPEN(QPRINT)
C                   OPEN      QPRINT
C                   ENDIF
C* Print the error and the API that received the error
C                   EXCEPT    BAD_NEWS
C                   EXSR      DONE
C                   ENDSR
C* DONE subroutine
C     DONE          BEGSR
C                   WRITE     TTLSEP
C                   WRITE     TTLS
C                   EVAL      *INLR = '1'
C                   RETURN
C                   ENDSR
C* WRTHEAD subroutine
C     wrthead       begsr
C                   WRITE     AFT1
C                   WRITE     AFT2
C                   WRITE     AFT3
C                   WRITE     AFT4
C                   ENDSR
C* DODETAIL subroutine
C     DODETAIL      BEGSR
C                   IF        *IN90 = *ON
C                   EXSR      WRTHEAD
C                   EVAL      *IN90 = *OFF
C                   EVAL      lstlib = *blanks
C                   END
C*
C                   IF        LSTLIB <> LIBN
C                   WRITE     AFMBR
C                   EVAL      LSTLIB = LIBN
C                   GOTO      GETOUT
C                   END
C                   IF        LSTFIL <> FILN
C                   WRITE     AFNOLIB
C                   EVAL      LSTFIL = FILN
C                   goto      GETOUT
C                   END
C                   WRITE     AFNOFIL
```

```
C     GETOUT        TAG
C                   add       jrnsec        tjrnsec
C                   add       jrnsiz        tjrnsiz
C                   ENDSR
OQPRINT    E            BAD_NEWS       1
O                                            'Failed in API '
O                       APINAM
O                                            'with error '
O                       QUSEI
```

D.3 Externally described printer file: PFILRPT

```
A*%%***************************************************************************
A*%%TS  RD  20010115  123614  SMBAKER     REL-V4R4M0  5769-PW1
A*%%FI+1066010000000000000000000000000000000000000000000000000000000000
A*%%FI       0000000000000000000000000000000000000000000000000000000000
A*%%***************************************************************************
A          R AFT1
A*%%***************************************************************************
A*%%RI 00000
A*%%***************************************************************************
A                                      SKIPB(001)
A                                      SPACEA(001)
A                                     3
A                                      DATE(*YY)
A                                      EDTWRD('0 /  /    ')
A                                    86
A                                      'Journal size estimate'
A                                   170
A                                      'Page: '
A                                     +0
A                                      PAGNBR
A*%%***************************************************************************
A*%%SS
A*%%CL 001
A*%%***************************************************************************
A          R AFT2
A*%%***************************************************************************
A*%%RI 00000
A*%%***************************************************************************
A                                      SPACEB(001)
A                                     80
A                                      '---------Record---------'
A                                   107
A                                      '--------Member--------'
A                                   131
A                                      '-----Journal Estimate-----'
A*%%***************************************************************************
A*%%SS
A*%%***************************************************************************
A          R AFT3
A*%%***************************************************************************
A*%%RI 00000
A*%%***************************************************************************
A                                      SPACEB(001)
A                                     42
A                                      'Record'
A                                     +3
A                                      '----------Journal----------'
A                                     82
A                                      'Number of'
A                                     97
A                                      'Average'
A                                   107
A                                      'Number of'
A                                   122
A                                      'Average'
A                                   131
A                                      'Average ops'
A                                   151
A                                      'MB per'
A*%%***************************************************************************
A*%%SS
A*%%***************************************************************************
A          R AFT4
A*%%***************************************************************************
```

```
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A                                     SPACEA(001)
A                                    3
A                                     'Library'
A                                    16
A                                     'File'
A                                    29
A                                     'Member'
A                                    42
A                                     'Length'
A                                    +3
A                                     'Name'
A                                    +8
A                                     'Library'
A                                    +5
A                                     'Act'
A                                    81
A                                     'Operations'
A                                    94
A                                     'per second'
A                                   107
A                                     'Operations'
A                                   119
A                                     'per second'
A                                   132
A                                     'per second'
A                                   154
A                                     'day'
A*%%*************************************************************************
A*%%SS
A*%%CL 001
A*%%*************************************************************************
A          R AFMBR
A*%%*************************************************************************
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A            LIBN          10A  O     3
A            FILN          10A  O    16
A            MBRN          10A  O    29
A            RCDLEN         5S 0O    43
A                                     EDTCDE(3)
A            JRNNAM        10A  O    +3
A            JRNLIB        10A  O    +2
A            JRNACT         1A  O    +3
A            RCDOPS        11S 0O    80
A                                     EDTCDE(3)
A            AVRRCDOPS      7S 1O    96
A                                     EDTCDE(3)
A            MBROPS        10S 0O   107
A                                     EDTCDE(3)
A            AVRMBROPS      7S 1O   121
A                                     EDTCDE(3)
A            JRNSEC        11S 1O   131
A                                     EDTCDE(3)
A            JRNSIZ        10S 3O   146
A                                     EDTCDE(3)
A*%%*************************************************************************
A*%%SS
A*%%SN JRNACT     x
A*%%*************************************************************************
A          R AFNOLIB
A*%%*************************************************************************
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A            FILN          10A  O    16
A            MBRN          10A  O    29
A            RCDLEN         5S 0O    43
A                                     EDTCDE(Z)
A            JRNNAM        10A  O    +3
A            JRNLIB        10A  O    +2
A            JRNACT         1A  O    +3
A            RCDOPS        11S 0O    80
A                                     EDTCDE(3)
A            AVRRCDOPS      7S 1O    96
```

```
A                                     EDTCDE(3)
A            MBROPS        10S 0O   107
A                                     EDTCDE(3)
A            AVRMBROPS      7S 1O   121
A                                     EDTCDE(3)
A            JRNSEC        11S 1O   131
A                                     EDTCDE(3)
A            JRNSIZ        10S 3O   146
A                                     EDTCDE(3)
A*%%*************************************************************************
A*%%SS
A*%%SN JRNACT     x
A*%%*************************************************************************
A          R AFNOFIL
A*%%*************************************************************************
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A            MBRN          10A  O    29
A            RCDLEN         5S 0O    43
A                                     EDTCDE(Z)
A            JRNNAM        10A  O    +3
A            JRNLIB        10A  O    +2
A            JRNACT         1A  O    +3
A            RCDOPS        11S 0O    80
A                                     EDTCDE(3)
A            AVRRCDOPS      7S 1O    96
A                                     EDTCDE(3)
A            MBROPS        10S 0O   107
A                                     EDTCDE(3)
A            AVRMBROPS      7S 1O   121
A                                     EDTCDE(3)
A            JRNSEC        11S 1O   131
A                                     EDTCDE(3)
A            JRNSIZ        10S 3O   146
A                                     EDTCDE(3)
A*%%*************************************************************************
A*%%SS
A*%%SN JRNACT     x
A*%%*************************************************************************
A          R TTLSEP
A*%%*************************************************************************
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A                                  129
A                                     '---------------'
A                                  146
A                                     '-----------'
A*%%*************************************************************************
A*%%SS
A*%%*************************************************************************
A          R TTLS
A*%%*************************************************************************
A*%%RI 00000
A*%%*************************************************************************
A                                     SPACEB(001)
A            TJRNSEC       11S 1O   131
A                                     EDTCDE(3)
A            TJRNSIZ       10S 3O   146
A                                     EDTCDE(3)
A*%%*************************************************************************
A*%%SS
A*%%CP+999CRTPRTF
A*%%CP+    FILE(ESTJRNSIZ/PFILRPT)
A*%%CP+    DEVTYPE(*SCS)
A*%%CP     PAGESIZE(*N       192      *N     )
A*%%CS+999CRTPRTF
A*%%CS+    FILE(QTEMP/QPRDRPT   )
A*%%CS+    DEVTYPE(*SCS)
A*%%CS     PAGESIZE(*N       132      *N     )
A*%%*************************************************************************
```

Appendix E. Comparing availability options

This appendix can help you compare availability options so that you can make decisions about what to protect and how. Journaling, mirroring, and device parity protection are compared by the extent of data loss, recovery time, and performance impact. Recovery time by failure type, and availability options by failure type are identified.

E.1 Journaling, mirroring, and device parity protection

Table 4 compares several important attributes of journaling physical files, mirrored protection, and device parity protection, including how they affect performance, the extent of loss, and recovery time.

Table 4. Journaling physical files, mirrored protection, and device parity attribute comparison

Attribute	Physical file journaling	Mirrored protection	Device parity protection
Data loss after a single disk failure	Loss of file data is determined by currency of backup	None	None
Recovery time after a single disk unit failure	Potentially many hours	None to a few hours	None to a few hours
Performance impact	Minimal to significant	Minimal, except some read operations improve	Minimal, except restore operations degrade

E.2 Availability options by time to recover

Table 5 shows which availability options can reduce the time needed to recover from a failure. The number of plus (+) signs in a column indicates an option's impact on the time to recover compared to the other options. An option with more pluses has greater relative impact.

Table 5. Availability options by time to recover

Option	DASD	System	Power loss	Program failure	Site loss
Save operation	+	+	+	+	+
Journal files	++	++	++	+	
Access path protection	++	++	++		
UPS			+++		
User ASPs	++				
Device parity protection	+++				
Mirrored protection	+++				
Dual systems	+++	+			++

Appendix F. Cost components of a business case

Creating an accurate business case for some IT applications is not trivial. This is certainly the case when justifying a high availability solution. Many of the benefits provided by high availability are intangible.

To help you create a business case for improving the availability of an application, this appendix provides a list of costs (both for providing availability and those associated with outages) that can be used as part of that business case.

Without a detailed study of your business, it is difficult to know whether an outage in your company will have a similar impact. Use this appendix as a guideline to justify the need for further study.

F.1 Costs of availability

The costs for providing an improvement in high availability are very intangible. The value of availability is much harder to ascertain. One of the first steps is to study the current availability statistics and understand which objective to improve. It is conceivable that a single faulty component, such as a local display, creates an invalid perception that the availability problem is within an application.

Review sources of information, such as problem management reports, system logs, operator logs, and so on, to identify the outages over the past year. Verify this list with the application owner to ensure that you both have the same perception of the current availability.

Next, identify and categorize the root cause for every outage, both planned and unplanned. From this list, identify the items with the highest impact on availability.

It is only when you understand what is causing the application to be unavailable at a given moment that you can effectively create a plan to improve that area. Use this information to identify which causes of outages must be addressed to gain the availability improvement you desire.

The plan is likely to include some change in processes, and it may also involve hardware and software changes.

The following sections provide information on some of the contributing factors for costs.

F.1.1 Hardware costs

A component failure impact analysis can be done to identify the single point of failure and the components that, if lost, would have a serious impact on the application availability. The only way to provide continuous operations is to have redundancy for all critical components.

Take the result of that study and discuss the results with the sponsors. There may be some identified components that are addressed as an expected upgrade process. Size and price the remaining components. At a minimum, consider the console hardware component.

F.1.2 Software

Just as redundant hardware is required to provide for continuous operations, redundant software is also required. Additional licenses of some programs may be required.

Does the current application need to be updated to support the high availability solution? Is this the time to add this application to help manage operations and availability? Are additional licenses required? When evaluating costs, consider:

- Application change control
- Change and problem management
- Utilities

F.2 Value of availability

The value of availability (or the cost of unavailability) is more difficult than arriving at the cost or providing that availability. For example, if you lose a network controller, what is the cost impact of the loss? This depends on many things:

- Which applications do the users in the affected area use?

  If the application developers access a test system, the cost will be lower.
- Which shift did the outage occur? What time?
- How long did it take to recover?
- How do you report availability?

F.2.1 Lost business

The amount of business lost because of an outage can vary from individual transactions, to the actual loss of a customer.

If the amount of business transacted by your application is consistent, compare the average value of business with the amount of business transacted on the day an application outage occurred.

It is difficult to tie the loss of a customer to an application outage. If you have a relatively small number of high-value customers, you probably have a close relationship with them and they may make you aware of why they moved their business elsewhere.

If you are in the retail industry, it is unlikely that you can produce a definite figure for the number of customers lost due to application unavailability. One possible method, however, is to follow a series of application outages and determine if it was followed by a trend in the amount of repeat business.

Either of these analysis methods require working with the application owner to obtain and record the required information.

It is likely that a single outage may result in lost transactions, where a series of outages may result in lost customers. Therefore, the cost of each outage is also impacted by the frequency of outages.

F.3 Image and publicity

As businesses become more computerized and visible on the Internet, electronic links between supplies and customers are becoming a standard. The availability of applications becomes more visible to those outside the company. With a click of the mouse, customers can go on to another source for their goods or services.

Recurring application outages are known quickly by customers (existing and potential). This impacts your validity for winning new contracts or renewing existing ones. Poor availability leads to bad publicity, which is very difficult to rectify.

To make matters worse, potential customers can be anywhere in the world. You should modify your view of the outage impact to reflect this.

It is nearly impossible to assess the cost of poor publicity caused by poor availability. If you have a public relations department, they may be aware of existing negative publicity and should have some idea of the cost to improve the public perception of the company.

F.4 Fines and penalties

Fines and penalties imposed as a result of application unavailability is an objective number to obtain. In some industries, application availability is monitored by controlling bodies. Companies are expected to maintain a certain level of availability, for example, the airline industry. There are also moves within the financial world to encourage companies to maintain high levels of availability. Extended outages can lead to fines from the governing bodies.

F.5 Staff costs

There may be significant staff costs both during and after an outage. Depending on the application affected, prolonged outages can have a significant financial cost in lost productivity. For example, for an application controlling a factory production line, there could be many people sitting around not able to do their job. Overtime may need to be paid to catch up on target productivity.

Identify the users of a given application, estimate the impact of an application outage on those people, and then multiply by their average salary per hour to provide a rough idea of the cost in terms of lost productivity. Factor in the overtime rate if lost productivity is made up by overtime to give a rough recovery cost.

Add the cost of the IT staff involved in recovering from the outage, and factor in any additional availability hours that may be required by the end users to catch up on their work.

F.6 Impact on business decisions

Depending on the type of application, the cost of an application outage varies considerably. This depends on how timely the information must be and the type of data involved. The loss of a business intelligence application varies from very

small to very large. For order processing in a factory working on just-in-time, the impact can be significant. If items are not ordered in time, the whole factory could halt production due to the lack of one key component.

Work with the application owners to identify the impact of the application unavailability for a given amount of time. Identify both a worst and best case scenario, and cost of reach. Then, agree on realistic average cost.

F.7 Source of information

To obtain some data to help in these calculations, check these sources:

- **Application**: If a business case was created for the application when it was first developed, there may be some cost benefit estimates in that document that are useful. Factor in the age of the document, however, to determine the applicability of the figures.
- **Disaster recovery**: If your company has a disaster recovery agreement, it is likely that a business case had to be presented in relation to that expenditure. It should contain estimates of the impact of a system outage. Additionally, when the business case includes the application or system level, use those figures or extrapolate an idea of the financial cost of the loss of a given application.

  Speak to the developers of the disaster recovery business case to see where they obtained the financial impact information used to build the case.
- **Transaction values**: Transactions for an average day can be recorded and used to assess the impact of an outage for an application. Record the number of transactions per day for each application. You can at least identify the change in the number of transactions if an outage occurs. If you can agree on an average value per transaction, this allows you to more readily estimate the financial impact of lost transactions.
- **Industry surveys**: Data processing firms and consultants have produced studies over the years, with example costs for outages. Their reports can include a cost range per hour and an average cost per hour.

F.8 Summary

Every industry and every company has unique costs and requirements. Tangible outage costs are only one part of the equation. You may be fortunate enough to suffer no unplanned outages, and yet still require better application availability. To maintain competitiveness, if your competitors are offering 24 x 7 online service, you may have no choice but to move in that direction. This may be the case even if there is currently not enough business outside the normal business hours to justify the change by itself. Additional availability can give you access to a set of customers you currently do not address (for example, people on shift work who do their business in the middle of the night). Internet shopping introduces a completely new pattern of consumer behavior. Previously, once a customer was in your shop, they were more inclined to wait ten minutes for the system to come back than to get back in the car and drive even minutes to your competitor. Now shopping is done by clicking a computer mouse. If you have better availability than your competitors, you have the opportunity to pick up customers from competing sites while their system is down.

Some retail outlets switch to a credit authorization business when the first provider experiences any interruption. They switch to another provider once the next interruption happens. If your systems have better availability, you have the opportunity to pick up a competitor's business when they experience an outage.

If your customer support systems are available 24 x 7, you have flexibility to have fewer call center staff. Once your customers realize they can get service any time, the trend tend to favor fewer calls during the day with more in the off-peak hours. This allows you to reduce the number of operators required to answer the volume of calls at a given time.

If you can spread the workload associated with serving your customers over a longer period of time, the peak processing power required to service that workload decreases. This can defer an additional expense to upgrade your system.

Due to mergers or business growth, you may be required to support multiple time zones. As the number of zones you support increases, the number and duration of acceptable outage times rapidly decreases.

A successful full business case includes these considerations, and others that are more specific to your company and circumstances. Most importantly, a successful availability project requires total commitment on the part of management and staff to work together towards a common goal.

Appendix G. End-to-end checklist

This chapter provides a guide to the tasks and considerations needed when planning a new high availability solution. It is not a definitive list and looks different for each customer, depending on the particular customer situation and their business requirements. Use it as a guide to help you consider factors influencing the success of a high availability solution.

Note: A service offering is available from IBM to examine and recommend improvements for availability. Contact your IBM marketing representative for further information.

G.1 Business plan

The investment in a high availability solution is considerable. It is critical that this investment is reflected back to the Business Plan. This will ease the justification of the solution, and in the process will display the value of the information technology solution to the business.

Does a valid business plan exist?

- Tactical Plan

  Do you have a tactical plan?

- Strategic Plan

  Do you have a strategic plan?

G.1.1 Business operating hours

Define the current operating hours of all parts of the business, no matter how insignificant. Suppliers and customers should also be included in this information.

How long can the customer business survive in the extent of a systems failure? Describe the survivability of the various different parts of the business, and rank their criticality.

- **Business operating hours**:
  - Current normal operating hours
  - Operating hours by application/geography
  - Planned extensions to operating hours
  - Extensions by application/geography
- **Business processes**:

  Do business processes exist for the following areas:

  - Information Systems Standards
  - Geographic standards
  - Centralized or local support
  - Language
  - Help desk
  - Operating systems
  - Applications

G.2 High availability project planning

Major points for a successful project plan are:

- **Objective of the project**: Prepare an accurate and simple definition of the project and its goals.
- **Scope of the project**: Define the scope of the project. This definition will more than likely be broken down to several major sub-projects.
- **Resources**: Are there sufficient resources to manage and facilitate the project?
- **Sponsorship**: Is there an Executive Management sponsor for the project?
- **Communication**:
  - Does the project have an effective communications media for the project?
  - Communications to the business, sponsors, and customers
  - Communications and collaboration of parties working on the project?
- **Cost management**:
  - Is there a budget for the project?
  - Has the budget been established based on cost of outage?

G.3 Resources

Soft resources are a critical success factor.

- **Current skills**: Do you have an accurate skills list? Do you have job specifications for the current skills?
- **Required skills**: What skills are required for the new operation?
- **Critical skills**: Do all these skills exist? Do you have a job specification for all critical skills?
- **Skills retention**:
  - Vitality

    Are existing resources encouraged to maintain their skills currency? Is there a skills development plan for existing resources?
  - Loss from the department
    - Are you likely to lose resources as result of the project?
    - Are these critical skills?
    - Through lack of interest in new mode of operation?
    - Through learning skills valuable outside the business?

G.4 Facilities

Are the facilities listed in this sections available?

- Testing the recovery plan
  - Do you have tested recovery plan?
  - Are there any activities planned during the project that could impact it
- Types of planned and unplanned activities?

G.4.1 Power supply

Ask the following questions about your power supply:

- **Reliability**:
  - Is the local power supply reliable?
  - How many outages in the past year?
  - How many weather related?
- **Quality**: Is the local power supply company committed to maintaining a quality service?
- **Power switchover**: Do you require power switchover?
- **UPS**:
  - Are there any UPSs in the installation?
  - Do they qualify with the particular power supply options?
  - Do they have the capacity to meet the new demands?
- **Servers**:
  - Is a UPS required for the Servers?
  - How many?
  - What type (battery/generator)/size/duration of backup supply/location?
  - Are the systems aware of power failure?
  - Do programs need to be developed to allow the systems to reaction to power failures?
- **Clients**:
  - Do clients need to have UPS?
  - What type (battery/generator)/size/duration of backup supply/location?
- **Other equipment**:
  - Is there other equipment that needs UPS?
  - Consoles/Printers/PABX/Network Devices/Machine facilities?
- **Generated supply**:
  - Will you be running a generated backup?
  - What configuration will be running?
  - Local power - generated backup
  - Generated power - local backup

G.4.2 Machine rooms

Ask the following questions regarding machine rooms.

- **Flooring**:
  - What is the current standard for your machine room?
  - Fully accessible or semi-accessible
- **Fire Protection**:
  - What is the current fire protection method?
  - Halon/Water/CO2
- **Air-conditioning**:
  - Is the machine room air-conditioned?
  - At what point will the equipment fail without air-conditioning?
  - Will the machine room ever achieve this condition with-out air-conditioning?

– Is there spare capacity in the air-conditioning system?
– Is there redundancy in the current air-conditioning?

- **Contracts**:
  – Do contracts and services levels exist for machine rooms?
  – Do these need amending for the new system?

G.4.3 Office building

Ask the following questions regarding your office building:

- **Workstations**:
  – Have the workstations been assessed for their ergonomic function?
  – Will the changes to the current system effect your liability for your employees ergonomics?
- **Cabling systems**:
  – Does the facility have a structured cabling system?
  – Will the existing cabling system accommodate the new system?
- **PABX**:
  – Is the current PABX capable of operating in the new environment?
  – Does the existing PABX have redundancy?
  – Is redundancy in the PABX a requirement?

G.4.4 Multiple sites

Answer the following questions regarding multiple sites.

- Are there multiple local sites (within the same site)?
- Are any of the local sites located across a public space? (highway, area)
- Are any remote sites less than 2km away?
- Are any remote greater than 2km away?
- Are any sites across an international border?
- Do the remote sites have a different PTT?

G.5 Security

In this section, answer the following questions regarding security:

- **Policy**:
  – Does a detailed security policy exist?
  – Is there a site security function?
  – Does this function set policy for I/T security?
- **Physical security**:
  – Is access protection provided?
  – Is there a documented process for dismissal employees?
  – Does this process also limit risk to systems prior to dismissal?
- **Office space**: Is the office space protected?

- **Machine room**:
  - Is the machine room protected?
  - Does the protection system provide an audit trail?
- **Site**:
  - Does the site have physical security?
  - Will the new system require this to be updated?
- **Operating system security**:
  - Do the systems implement security?
  - What level is implemented?
  - What level is required?
  - What applications at what level?
- **AS/400 security levels**:
  - What level of AS/400 security is implemented?
  - What level is required?
- **Personal computer security**:
  - What forms of PC security have been implemented?
  - Do these levels need changing?
- **Remote users**:
  - What type of remote security has been implemented?
  - How do remote users access the systems?
  - Intranet?
  - Internet?
- **Network security**: What network security is in place?
- **Printing**:
  - Are there secure printing requirements?
  - Are the printers producing secure output located in a controlled environment?

G.6 Systems in current use

Document the model, processor feature, interactive feature, main storage, DASD capacity, DASD arms, protection, and ASP.

G.6.1 Hardware inventory

Prepare the inventory list of the all the hardware components in your current system, including:

- Processors
- DASD subsystems: DASD Space available
- Mirroring
- RAID-5
- Third Party solutions

- Tape subsystems
  - Multiple tapes subsystems versus tape library
  - Data transfer rate
- IOPs and Towers

G.6.2 Redundancy

What level on redundancy are you planning?

- System (cluster)
- Tower
- Bus
- I/O processor
- Controller
- Adapter

G.6.3 LPAR

- Are you planning to use LPAR support for the project?
- Transition support with LPAR?
- Server consolidation support with LPAR?

G.6.4 Backup strategy

Even with a continuously available solution backups are necessary. Is there a new backup strategy in plan?

Does this plan include the following components?

- Media Management
- Media storage
- Automated backup
- Performance
- Save-while-active
- Incremental backups
- Disaster recovery backup

G.6.5 Operating systems version by system in use

- Interactive CPU, batch CPU, DASD utilization, network utilization LAN/WAN.
- Are there multiple operating systems versions in this plan?
- Do you plan to rationalize these into the same version?
- Is there a plan for bringing systems to the same version

G.6.6 Operating system maintenance

- **Servers**: Do you have maintenance contracts for your server operating systems?
- **Clients**: Do you have maintenance agreements in place for your client software?

G.6.7 Printers

- Do you have a list of your printer inventory?
- Are the printers supported with a maintenance agreement?
- Will the printers support the new environment?

G.7 Applications in current use

Inventory of server based applications

- Application provider
- Version
- Number of users
- Database size in MB
- Data transfer requirements

Inventory of client based applications

- Application provider
- Version
- Number of users
- Database size in MB
- Data transfer requirements

Application maintenance

- Support contracts
  - Do support contract exist for all applications?
  - What is the support level?
  - Are there react time guarantees?
  - Does this meet the new availability requirements?
- Frequency of update
  - Does the application have updates?
  - What is the frequency of update?

G.7.1 Application operational hours current

Ask youself the following questions:

- By application what are the operational hours?
- Are there any processing peak periods that extend these operating hours? Application processing peaks.

Application processing map

An application processing map shows the time plan of when various applications run on both server and client machines. It shows the interleaving of applications and should identify any periods that have an opportunity for very high or very low requirement.

- Interactive
- Batch
- During the day
- Day end
- Month end
- quarter end
- year end
- fiscal year end

Application operational hours requirement: 5x8, 7x24, 24x365

What are the assessed operational requirements of each application?

Growth information; applications, database, users
What is the growth information for each application and its associated database and users?

Splitting an application
Is the splitting of the application across multiple machines a consideration?

Appendix H. Special notices

This publication is intended to help customers, business partners, and IBMers who are looking to implement a high availability solution for their AS/400 system. It provides planning checklists and background information to understand the facets of planning for a high availability solution, implementing a high availability solution, and managing the project installation itself. 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 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, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating

environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)®
IBM ®
APPN
AS/400e
CT
DataJoiner
DB2
Netfinity
OS/2
RPG/400
SP
System/38
400
Lotus Notes
Tivoli

Redbooks
Redbooks Logo 
AS/400
AT
Current
DataPropagator
DRDA
Nways
OS/400
SAA
System/36
XT
Lotus
Notes

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries. (For a complete list of Intel trademarks, see http://www.intel.com/tradmarx.htm)

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.
Other company, product, and service names may be trademarks or service marks of others.