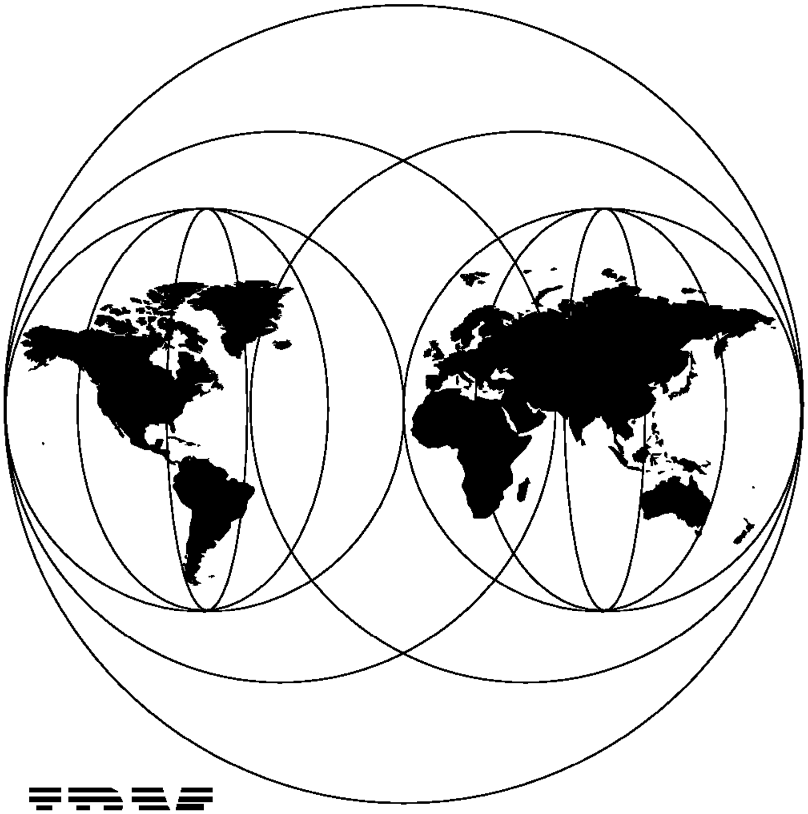
SG24-4735-00

AS/400 Performance Management V3R6/V3R7

March 1997



International Technical Support Organization
Rochester Center

International Technical Support Organization

SG24-4735-00

AS/400 Performance Management V3R6/V3R7

March 1997

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 429.

First Edition (March 1997)

This edition applies to OS/400 Version 3 Release 6 (5716-SS1), OS/400 Version 3 Release 7 (5716-SS1) and Performance Tools/400 (5716-PT1).

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

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

Contents

Preface

This redbook describes a methodology for performance management on IBM AS/400 Advanced Systems using PowerPC technology with OS/400 Version 3 Release 6 and OS/400 Version 3 Release 7. It includes setting up performance objectives, collecting and reviewing performance data, tuning of resources, and capacity planning. Performance guidelines and application design tips are also provided.

This document is intended primarily for IBM and IBM Business Partner technical professionals who want to implement a performance management structure on an AS/400 customer system. It is also useful for IBM customers who have in-depth AS/400 technical skills.

An intermediate knowledge of Performance Tools/400 Version 3 Release 6 and Performance Tools/400 Version 3 Release 7 (licensed program 5716-PT1) is assumed.

The Team That Wrote This Redbook

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

Neil Willis is a Senior ITSO Specialist at the International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of AS/400 performance and capacity planning. Before joining the ITSO one year ago, Neil worked in AS/400 System Engineering in Australia as an Advisory Systems Specialist.

Rick Turner is recognized worldwide for his work in writing, teaching, and consulting on AS/400 performance. He has been involved in performance work since the early days of the System/38 and has been active in performance work since 1980. He retired from IBM in mid-1993 after 29 years with IBM. He currently writes for NEWS/400 magazine and has co-authored a second book on AS/400 performance published by Duke Press.

Debbie Hatt is an AS/400 System Specialist in the UK. She has 14 years of experience in application development on the AS/400 platform and S/38 platform. She holds a degree in Mathematics, Statistics, and Operations Research from Westminster University in the U.K. Her areas of expertise include application development, specializing in system and application performance.

Ian McClintock is a Software Service Specialist in Australia. He has 23 years experience working for IBM in customer engineering and product software support. In addition to performance and capacity planning, Ian′s other areas of expertise include software problem analysis and AS/400 availability/recovery.

Alberto Otero is an AS/400 Systems Engineer in Chile. He has 22 years of experience in the IBM software field on the AS/400 system, S/36, VM, and MVS platforms. He holds a degree in Civil Engineering from Universidad Catolica de Chile. His areas of expertise include AS/400 performance and capacity planning specializing in system and application performance. He has written extensively and teaches IBM classes in the IBM Education Center in Santiago de Chile.

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

Jim Cook, International Technical Support Organization, Rochester

Dick Bains, IBM Rochester
Sue Baker, IBM Rochester
Larry Cravens, IBM Rochester
Jim Denton, IBM Rochester
Charlie Farrell, IBM Rochester
Kim Greene, IBM Rochester
Allan Johnson, IBM Rochester
Dave Johnson, IBM Rochester
Jay Kurtz, IBM Rochester
Mahdad Majd, IBM Rochester
Bettina Martin, IBM Germany
Cathy Mestad, IBM Rochester
Brad Nelson, IBM Rochester
Ricky Peterson, IBM Rochester
Lloyd Perera, IBM Australia
Paul Remtema, IBM Rochester
Jeff Ryan, IBM Rochester
Diane Straka, IBM Rochester
Bill Teo, IBM Rochester
Richard Theis, IBM E&T, Paris
Don Ward, IBM Rochester
Dale Weber, IBM Rochester

Comments Welcome

Your comments are important to us!

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

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 443 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

For Internet users	`http://www.redbooks.ibm.com`
For IBM Intranet users	`http://w3.itso.ibm.com/redbooks`

- Send us a note at the following address:

  `redbook@vnet.ibm.com`

Chapter 1. Introduction to Performance Management

Performance management is a strategy for planning, implementing, and controlling all tasks a customer has to perform to measure and achieve acceptable performance including capacity planning, collection of performance data, and tuning of resources.

1.1 Why Do Performance Management?

Performance management on a computer system is as necessary as maintaining an automobile or balancing a checkbook. If you use the automobile very little or never write checks, you have almost no maintenance requirement. Similarly, if you have a system that has few transactions or records to process, performance management actions may be ignored.

However, if you write checks, you should perform reasonable maintenance activities such as deducting checks and adding deposits. If you drive your auto, you should check oil and water level, gasoline level, and tire air pressure. Based on guidelines, you add one of the fluids or air when necessary.

Oil sticks provide marks indicating oil level. By checking this mark on a regular basis, you can detect the progress (trend) toward the ″add oil″ indicator. When the oil level reaches the ″add guideline″, you know you should add oil.

If you do not examine your checkbook or fluid level on a regular basis, you may miss the trend toward insufficient funds, or low fluid levels, or may not detect signs (for example, an oil leak) of potential problems. You are increasing your chances of having an ″unplanned interruption″ to your normal activities. Instead of detecting and repairing a minor problem according to a planned schedule, a major problem may occur at the worst possible time. Recovering from some major failure may require several hours or days of effort. If you write checks with insufficient funds you may have your credit rating lowered, or you may not be able to purchase necessities. Even if you have an ″automatic loan″ account, continued neglect of your balance can lead to serious financial problems. If your auto breaks down, you may be far away from any assistance or, even worse, you may have a serious accident. In some cases, your planned activities may become suspended for several hours or days.

Performance management is necessary to optimize utilization of your computer system by measuring current capabilities, recognizing trends, and making appropriate adjustments to satisfy end user and management requirements such as response time or job throughput. It is needed to maintain business efficiency and avoid prolonged suspension of normal business activities.

Saving previously collected data is necessary in determining trends and possible trouble areas. For example, it is common to hear ″my response time or batch job run time has degraded and there have been no changes to the system″. Without historical data, it is more difficult to identify areas to investigate. In many cases, comparing historical data identifies such trends as degrading response time, an increasing number of records (″disk I/Os″) being processed, or increased CPU utilization before performance becomes a visible problem.

Lack of performance management, especially when a new release is installed or when a new application has been added or workload has increased (for example, workstations are added or more records are being processed) can lead to a severe business impact. This can be avoided and growth can be planned for if proper performance management is part of normal data processing activities.

You should always collect and save Performance Monitor data before making a change such as installing a new release, adding a large number of active work stations, or adding a new application.

1.2 Performance Management Methodology

This document identifies the following components of performance management:

- Setting performance objectives
- Performance measurement
- Performance data analysis
- Performance trend analysis
- Performance problem analysis

The basic idea is to use currently available tools and techniques combined in an approach that (almost) automatically keeps track of performance. For example, a lot of performance measurement can be done with the Performance Monitor shipped with every AS/400 system.

We decided to use the Performance Tools/400 Licensed Program (5716-PT1) as the basis for data and problem analysis. While this licensed program is the principal tool used, additional tools, including internal IBM tools, are discussed when they provide information or function not available in 5716-PT1.

AS/400 query products can be used to process the Performance Monitor data in ways that are not directly available through formal performance. For example, queries can be run to compare or summarize the content of different sets (collections) of Performance Monitor data or produce reports from Performance Monitor database files not included in Performance Tools/400 report output.

Many tools used in this approach can be packaged into a procedure, or better yet a Control Language (CL) program, that is automatically started at certain times.

With an automated technique, you can guarantee having performance data available even when there is no performance problem and there is a need to extrapolate the requirements for future growth *before a problem occurs that impacts the current business operation*.

Having a long history of gathered performance data also makes it easier to detect peaks in the workload on a longer term, such as month end processing, fiscal year end, and other seasonal events.

1.3 Benefits of Using the Performance Management Methodology

Benefits for the AS/400 Customer

The following list contains some of the things that AS/400 customers can benefit from by using the performance management methodology:

- Allows a more stable level of performance by measuring performance data against objectives and controlling resource utilization.
- Provides a knowledge base of the components of performance to better understand application function versus performance trade-offs.
- Provides a management planning vehicle in order to plan ahead for system upgrades, new applications, growth in number of users, and changes in workload.
- Establishes a base of historical data that proves the value of a centralized computer in terms of providing services to end users.

Benefits for IBM and Business Partners

The following list contains some of the things that both IBMers and business partners can benefit by using the performance management methodology:

- Having a customer who is satisfied with system performance and understands the realities of application function versus performance.
- Having measured data that accurately depicts the customer performance and movement in the consumption of system resources.
- Enables a focus on ordered system growth rather than reacting to an unanticipated performance crisis.

1.4 Prerequisites

Performance Skills. We assume the reader of this publication has some knowledge about AS/400 performance but cannot or has not formulated an ordered, disciplined use of AS/400 performance tools and analysis of the performance data. Therefore, we do not explain in great detail how the tools for performance work, but rather the order in which to use them, and how to package them together.

1.4.1 IBM Performance Tools

Performance Tools for OS/400 Version 3 Release 6, 5716-PT1, is a program product that provides a set of reporting, analysis, and modelling functions to assist an AS/400 user to manage the performance of the system. It provides printed and online reports, and these can be in graphic or tabular form. A performance advisor function assists the user in analyzing system performance and provides recommendations. Performance Tools for OS/400, through its modelling facility, can be used to help predict probable system performance before changes are made.

Performance Tools for OS/400 makes use of an easy-to-use menu interface. From this menu interface, users can initiate requests for performance reports, and use the collected data as input to a capacity planning model.

The performance advisor component of Performance Tools for OS/400, as well as making recommendations to improve system performance, also implements tuning recommendations, if specified by the user. The knowledge-based advisor also provides detailed explanations of its analysis, which is of great benefit to novice and experienced users.

A capacity planning product, the *BEST/1 Capacity Planner,* written by BGS Systems, is integrated into Performance Tools for OS/400.

Performance Tools for OS/400 is divided into three elements:

Enabler The enabler is the base code onto which you **must** add manager or agent.

Agent Adding agent to the enabler gives a subset of Performance Tools functionality for those customers who do not require all of the tools required with manager and enabler. Key functions such as collect performance data, delete/copy/convert data, display performance data, work with historical data, and the advisor are included in the agent. The agent feature supports the creation of a BEST/1 model from Performance Monitor data. This permits a remote location to send a model as a database file (QACYMDLS, member=modelname) to a central AS/400 location where the full functions of BEST/1 can be used for capacity planning.

Manager Adding manager to the enabler gives full Performance Tools functionality. In addition to the functions described in the agent, the manager includes performance reports, capacity planning (using BEST/1 Capacity Planner), programmer performance utilities, work with system activity, and performance graphics.

You can automate the collection of Performance Monitor data (Start and End Performance Monitor (STRPFRMON/ENDPFRMON) commands) either by using the OS/400 Job Scheduler function directly or using the Add or Change Performance Collection (ADDPFRCOL/CHGPFRCOL) commands.

The ADDPFRCOL/CHGPFRCOL command technique requires that a system job (QPFRCOL) is started that looks at the performance collection start values and starts and ends the Performance Monitor accordingly.

Either through job scheduling or the user exit program (EXITPGM) parameter on the STRPFRMON command, the following functions of the Performance Tools/400 can be automated:

- Various printed report options (PRTcccRPT commands)
- Creation of historical data (CRTHSTDTA command)
- Analysis of Performance Monitor data through the advisor function (ANZPFRDTA command)
- Creation of a BEST/1 model (CRTBESTMDL command)
- Analysis of a BEST/1 model (ANZBESTMDL command)

Performance Tools/400 for Version 3 Release 6, SC41-4340, and *BEST/1 Capacity Planning Tool for Version 3*, SC41-4341, describe the automation capabilities and the use of the performance tools.

Performance Tools for OS/400 Version 3 Release 6 includes the Performance Explorer, which can be used for detailed analysis of performance using three different trace techniques. These techniques are:

***PROFILE** This traces the high level instructions of a program. It is similar to the Sample Address Monitor found in Performance Tools for OS/400 Version 3 Release 1 and previous versions of OS/400.

***STATS** This provides a trace of module calls within an executing job. It is similar to the PRPQ **TPST**, which was available with previous releases of OS/400.

***TRACE** This option can collect low level data in a number of ways that can help resolve performance problems that cannot be readily resolved by other performance tools. By its nature, it is flexible, but can only trace for short periods of time due to the large amounts of data being collected and the additional workload that is placed on the system. A number of query programs can be used to analyze the collected trace data.

It is beyond the scope of this document to elaborate on the use of the Performance Explorer (PEX). Limited information is available in the *Performance Tools/400 Version 3 Release 6*, SC41-4340. Also, it is the intent of the ITSO in the near future, to publish a redbook dedicated to using the Performance Explorer.

Chapter 10, “Additional Performance Tools” summarizes several performance tools in addition to the Performance Tools/400 licensed program. The ability to perform query functions on performance monitor data is also discussed in that chapter. In Appendix C, “IBM Internal Use Only Tools/Documents,” you can find an overview of several AS/400 ″Internal Use″ or ″as is″ tools available from the IBM VM-based tools disks or from some individuals around the world.

A performance management service offering, Performance Management/400, is also available in most countries to provide trend analysis of system resources in graphical format. More information about this service is available in Section 10.2, “Performance Management/400 (PM/400)” on page 147.

1.4.2 Other Publications, References

We did not intend to reproduce formal publications containing AS/400 performance information such as the *Performance Tools/400 Guide*, SC41-4340, HONE performance items, or other ITSO documents. In some cases, we may summarize information contained in these sources and reference them for additional details.

We have included much information in this document from ″internal″ or ″informal″ sources, such as COMMON presentations and tools available on the AS4TOOLS disk, since much of this information has been of valuable assistance in certain customer situations and there is limited field awareness of this information.

See Appendix I, “Related Publications” section of this document for additional AS/400 performance documentation.

1.5 Future Performance Management Guidelines

AS/400 ″performance information″ exists in many formal and informal sources of information. This document attempts to collect this information into a central repository to facilitate the use of the information to educate appropriate personnel and establish an AS/400 environment focused on ordered performance management and customer satisfaction with performance.

We solicit comments, additional guidelines, and requests for areas of additional information through the reader′s comments form. This input can be used to determine future updates to this publication or additional performance-oriented ITSO publications.

Chapter 2. Performance Requirements and Objectives

One of the first tasks in a system performance/capacity study is to make sure everyone reaches a clear understanding of the performance goals as measured by the users. The system performance criteria should be clearly stated and well understood by everyone before doing the analysis and mainly before making any changes.

You may have a mix of applications with performance criteria that vary over time depending upon the currently active applications. When interactive and non-interactive jobs run concurrently, they may have conflicting performance requirements.

Both interactive and non-interactive processing objectives should be defined in terms that can be measured either automatically by the system or through monitoring by a person. An objective must also be reasonable based on the complexity of the processing and the power of the system model being used.

Note that ″batch-type″ processing is independent of the ″job type″ classification assigned by the system in some of the AS/400 performance reports. For example, a ″transaction″ that processes 1000 records before displaying data on a workstation may be part of an interactive ″job type″ from the performance monitor viewpoint, but this transaction is a batch (non-interactive) type function from a system resource utilization viewpoint.

It is important to keep the different processing requirements in mind. Often the analyst′s perception of adequate performance does not match the user′s view. That is why one of the analyst′s responsibilities is to ensure that a smooth transition occurs through the stages of definition, measurement, analysis, and follow-up during a system performance study.

Start the performance definition work by establishing specific performance criteria and system resource utilization guidelines to be used to evaluate system performance. The user may say that a certain set of non-interactive jobs must run every day at the close of business and they must run in one hour. That is a good objective from the user′s point of view, but there are other things to consider.

Keep in mind, the objective of a performance definition effort is to translate the user′s system performance needs into a well-defined statement of goals and requirements.

2.1 Criteria for Interactive Jobs

Considering interactive jobs, one definition of good performance is that response time is good enough to ensure that customers do not perceive abnormal delays. Other criteria may require end-of-day processing to be completed by a specific time.

Some suggested ways to specify interactive performance criteria include:

- Local users: n% transactions have response time less than x seconds.
- Remote users: n% transactions have response time less than x seconds.

- Interactive transaction rate should average x transactions per hour.

Even though the user can set certain response time objectives, some of the transactions might utilize the resources more than the response time objective allows. It should be noted that while the overall objectives should be based on average values, system sizing needs to consider the peak load conditions.

Refer to Chapter 3, “Factors Affecting Performance” on page 11 and Section 3.1, “Interactive Response Time Components” on page 11 for more details.

2.2 Non-Interactive (Batch) Criteria

Non-interactive work includes the typical batch job work as well as AS/400 spooled work, client/server work, work triggered by the Submit Job command, and work triggered by message queue and data queue entries.

In most cases, the performance objectives should state the amount of work performed and the amount of resources used in a specific amount of elapsed time. From the users point of view, the amount of work performed is a well-defined measurement of business volume such as number of credit card authorizations per hour, number of customer orders handled per shift, number of trucks arriving or leaving the back dock in a day, or other metric relative to the business. The amount of resources used to perform this work is equally important in that it defines the amount of CPU needed, the number of DASD arms, or the amount of DASD space needed. Both the amount of work and the amount of resources used need to be considered.

Non-interactive criteria is relative to a submit job command (job start), an incoming *program start request* (job start), a scheduled job start, a message appearing on a message queue, a data queue entry appearing on a data queue, or a new spooled file entry appearing on an output queue. A ″non-interactive transaction″ includes job run time, time to complete printing a spooled file, time to complete a set of related jobs, time to process ″n″ message queue or data queue entries, the number of records processed per hour, or time to send or receive a ″file″ or ″files″ between a server and its attached client (or clients). In the ADSTAR Distributed Storage Manager/400 (ADSM/400) environment, it includes the time taken to ″backup″ or ″restore″ files and directories between the AS/400 system and one or more client workstations.

In many cases, knowing the number of records processed per unit of time, the number of disk I/O operations per second, or the number of bytes (characters) transferred by these non-interactive jobs is the only way to set realistic expectations or evaluate degradation or improvement in ″run times.″ Many of the components of interactive work also apply to non-interactive work.

The customer should agree with those responsible for performance management about what defines the unit of measurement by which they judge performance to be acceptable or non-acceptable.

2.3 System Expectations

Once performance objectives are set, the next task is to determine hardware that can handle your workload and meet the objectives.

To assess your current hardware capabilities, you need to determine average disk use, average CPU use, and average memory use among other things.

2.4 User Expectations

Performance objectives of a specific customer might be quite different from the objectives of another customer. Objectives must be realistic, based on the type and number of operations performed and the base capabilities of the hardware involved. For example, running 3500 high-level language statements during an interactive transaction may not deliver sub-second response time on any current models of the AS/400 system. Sending full displays of data to 10 or more workstations on a 4800 bps (bits per second) remote communications line does not deliver sub-second response time, regardless of CPU instruction speed. The following examples illustrate what can and cannot be expected on the AS/400 system:

- Running through 500 statements of RPG or COBOL code and 20 or less database I/Os per transaction on a mid-sized AS/400 system probably gives you reasonable response times and good throughput in terms of transactions per hour.
- Performing 50 database input and output operations (DB I/Os), or running through 2000 RPG or COBOL instructions per transaction rarely results in consistent sub-second response times.
- Filling a subfile with 1000 records before the first display is shown does not yield good response time.
- Using 10 or more database files in a program and performing full (non-shared) file opens and closes during a single transaction results in, at best, erratic response time, if not unacceptable response times.
- Using token-ring speeds for interactive transactions or Distributed Data Management (DDM) I/O operations usually results in acceptable performance. In many Personal Computer (PC) workstation LAN environments, using any of the ″5250 emulation products″ (V2R3 DOS PC Support/400 Work Station Function (WSF), Windows and OS/2 RUMBA, or OS/2 Communications Manager 5250 Emulation) achieves response times equivalent to twinaxial attached workstations.
- When an interactive program does a few database operations and DDM is used to access remote data, response time can be acceptable. But as the number of program database operations increases, display station pass-through normally delivers better performance. This is because the display data is much less than the data including many database records.
- When two-phase commitment is used through either APPC programming or DRDB interfaces, expect performance delays when confirmation is being performed to remote systems.
- When multiple jobs are running, the priority of those jobs can impact the time it takes to get a job or transaction done. When the same application is run in all of the jobs of the same priority, all jobs, in general, are treated equally. However, if one of the jobs performs batch-type processing (large

number of CPU instructions or database I/O operations) at an equal or higher priority than the other jobs, that job may ″use more of the CPU″ and degrade the performance of those other jobs.

- For client/server environments, long running queries initiated from the client should see the same AS/400 server CPU utilization, disk utilization, and response time as local AS/400 interactive or batch jobs running the same query.
- For client/server environments, running online transaction processing transactions (short running queries) should see slower response times than local AS/400 interactive jobs.

  Performance can be improved by using:

  - SQL package support
  - Re-using previously prepared SQL statements with parameter markers (variables) for repetitive queries
  - Stored procedures (already created SQL programs)
- For client/server environments running with a slow processor client, expect slower performance compared to a high performance client.

Refer to Chapter 11, “System Performance Tuning Tips” and Chapter 12, “Design and Coding Tips” for more wide-ranging performance tips.

Chapter 3. Factors Affecting Performance

Many factors affect performance. Invariably users complain only when performance is bad or inconsistent. In many cases, the cause of poor performance will become quickly apparent, in other cases less obvious. It is important to understand the factors that affect performance, when to look for them, and how to measure their impact.

3.1 Interactive Response Time Components

The following chart illustrates the various components of response time. Each element can contribute a portion to total response time.

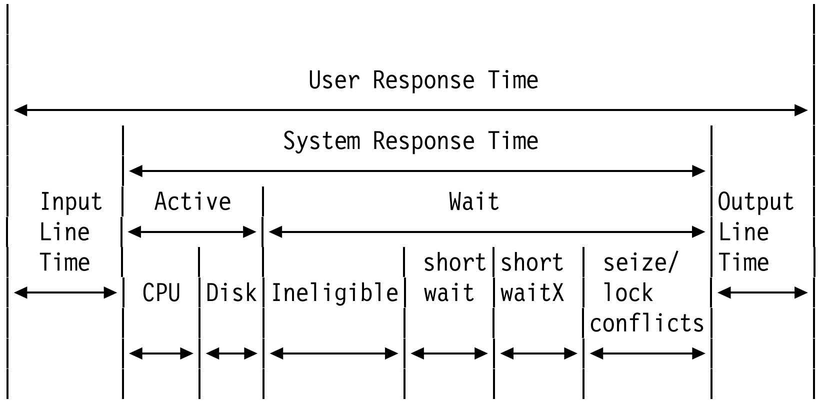


Figure 1. Components of Interactive Response Time

Each transaction uses some amount of communications line capacity, CPU time, main storage, does a number of disk accesses, and has to be scheduled for the CPU using a priority classification. The interactive response time experienced by a user is the total of many components:

- There is a transmission time delay for the transaction to reach the CPU. (This can be significant in some cases such as token-ring or remote workstation.)
- Once the transaction reaches the system, the system′s response time measurement begins.
  - The job may have to wait for an activity level at the system.
  - Once the activity level has been entered, resource utilization begins, which includes:
    - CPU processing time (including CPU queuing)
    - Disk I/O time (including disk queuing)
  - There are also periods of inactivity during which the transaction waits in any one of a number of states, including:
    - Ineligible time (not holding an activity level)
    - Short waits (still holding an activity level)
    - Short waits - extended (not holding an Activity Level)
    - Object or record seize wait conflict (still holding an activity level)
    - Object or record lock wait conflict (not holding an activity level)
- Finally, there is a transmission delay in the response reaching the user.

Note: If you are running pass-through or 3270 emulation, the output line time includes the target system service time. In a client/server environment (Distributed Data Management (DDM), Client Access/400, VRPG, LAN Server/400, translation between EBCDIC and ASCII, and so on), the processor speed and any other work being done concurrently on the remote system or workstation must also be considered.

In some environments, the transaction may encounter errors that need handling. In addition, the workload demands on the system can affect high-priority jobs. Refer to *Performance Tools/400*, SC41-4340, for details on the various components of response time.

Note: Input and output times for local twinaxial or token-ring local area network (LAN) communication lines are normally so small that they are not considered. However, in more and more cases, LANs are composed of local and remote LANs connected over some ″high-speed″ communication line and ″bridging″ or ″routing″ devices. In these ″remote LAN″ situations, interactive response time or non-interactive throughput is limited by the *slowest link* connecting the system with the workstation.

The OS/400 performance monitor can collect response time data from IBM 5494 remote workstation controllers. This information is reported in a range of ″response time buckets″ for dependent workstations attached to the IBM 5494 controller. Apart from this, **neither Performance Monitor data nor Performance Tools response time output includes communication line time**. Overall communication line information is collected by the Performance Monitor and the collected data can be processed by the Performance Tools or user-written queries over the QAPMxxx database files as described in *OS/400 Work Management Version 3*, SC41-4306. Refer to Section 11.12, “Communication Performance Considerations” for additional communication line considerations.

To analyze system performance and determine what changes should be made, you need to consider the following:

- Review comparative data between acceptable and unacceptable performance. Consider both ″normal″ and ″peak load″ situations. Without it, you cannot define the target values you need to achieve.

  Find the contribution of each component (see Figure 1 on page 11) of interactive user response time. Determine this for transactions that are both performing as they should and those that are not.

- Collect comparative data before and after changes are made.

  This enables you to evaluate the effect of the changes you have made.

- How do you identify which component is causing problems? It may be that one or more is out of range and needs attention.

- What can be done to reduce the effect of a measured bottleneck?

In some situations, the system may be working exactly the way it should, given the type of work being performed. System performance may degrade primarily because of workload, but it can also be affected by improper system tuning, less than optimum application design and program implementation, error recovery, or job scheduling.

When system performance is less than it should be, one or more of the components of job performance (Figure 1 on page 11) is involved. The challenge is to find out which is causing the most severe impact on performance, determine the effect of changing it, and determine the cost to correct it.

The solution to a problem may not be just adding hardware. In many cases, a change to the application or system setup can significantly improve performance. This is appropriate in that even greater performance gains may occur once the hardware is upgraded.

3.2 Basic Queuing Theory

Customer expectations for a single job or specific transaction must be balanced against realistic expectations when many jobs are active during the same period of time.

The work of a single job or transaction within that job is comprised of several tasks or services. The work given to a task is called a **request** or sometimes a unit-of-work. The task is also called a **server** and the time taken to complete processing the request is called the **service time**.

Note: A single server can service only one request at a time. Multiple requests wait for service by the server.

Using Figure 1 on page 11 as a reference, the servers of the components of response time include CPU time and disk I/O time. As Figure 1 on page 11 shows, there are wait times associated with these servers, including waiting for CPU and waiting for disk I/O. These wait times are associated with queuing for the server. The higher the server utilization, the greater the wait or queuing time.

Queuing is a concept that applies to computer resources just as it does to people waiting in line at the supermarket or waiting to access an Automated Teller Machine (ATM). In general, how long it takes to get a request or unit of work serviced, whether it be a request to complete the purchase at the supermarket counter, complete a cash withdrawal at the ATM, perform a disk I/O operation, or use the CPU depends on three primary parameters:

- The number of ″waiters″ in the line ahead of a new request
- The number of servers responding to requests
- The service time to complete a request once given to the server, which is a function of the speed of the server and the amount of work to do

There are mathematical equations to determine the effect of queuing and two of them are discussed in the following topics for disk and CPU. The formula for computing the queuing multiplier assumes:

- Work arrives in a normal (or Poisson) distribution pattern.
- Requests for the resources are not all for the same amount.

As the utilization of a server increases (more work for the server), queuing accounts for part of the longer work (or request) completion. In an interactive transaction, this can be considered the major cause of long response times. The Queuing Multiplier (QM) is a measure of the effect of queuing. Table 41 on page 386 shows the approximate QM based on CPU utilization values. Using a simple example, assume the CPU is 67% utilized. The mathematical equation says the QM for a single CPU is three. A QM of three means, on the average, there are a total of three requests in the queue (you and two requests for work ahead of you). Therefore, using an average of .2 seconds of CPU to service a request, an interactive transaction (response time) takes a minimum of .5 seconds to use the CPU (server response time = QM x stand-alone service time).

The components of response time show that CPU is only one of the resources (servers) involved in response time. Disk service time, which is a function of the disk utilization and the disk QM, also must be factored into response time expectations. In real environments, additional wait times such as exceptional wait times, also need to be factored into expectations. These exceptional wait times (waiting for record or object locks, waiting for communication line data transmission, and so on) can play an important part in actual performance results and must be included in analyzing performance problems and capacity planning.

The Queuing Multiplier is an important factor when projecting the impact of adding work or additional hardware on current system performance. Note that the Performance Tools Capacity Planning support assumes a reasonably well-tuned system that assumes a CPU QM of four or less. Systems with performance problems often show resources with higher Queuing Multiplier factors. When using the Capacity Planner with measured data, a QM of greater than four generates less accurate results.

The Performance Tools Transaction Report - Job Summary lists the CPU Queuing Multiplier calculated at each job priority for the collected data.

3.2.1 Queuing Multiplier Effect

The Queuing Multiplier values used in the formulas for disk and CPU service time can be shown graphically. The curve in Figure 2 on page 15 shows the utilization at various rates and the significance of the ″knee″. The knee of the curve is the point where a change in utilization produces a correspondingly higher change in the Queuing Multiplier. That is, the change along the Y-axis (Queuing Multiplier) is significantly greater than the change along the X-axis (utilization). The knee of this curve is the maximum utilization point that a certain resource should be driven up to. After this ″knee″, service time becomes less stable and may increase dramatically for small utilization increases.

Not all resources (servers) have the same effect on performance at the same utilization values. There are different recommended maximum values for the different resources, such as CPU, disk, memory, controller, remote line, IOPs, and so on.

Performance Tools/400 Version 3, SC41-4340, provides more information on queueing. Figure 2 on page 15 shows a simplified queuing formula and a curve derived from it. It highlights the effect of increasing utilization on the Queuing Multiplier for a single server:

Arrivals

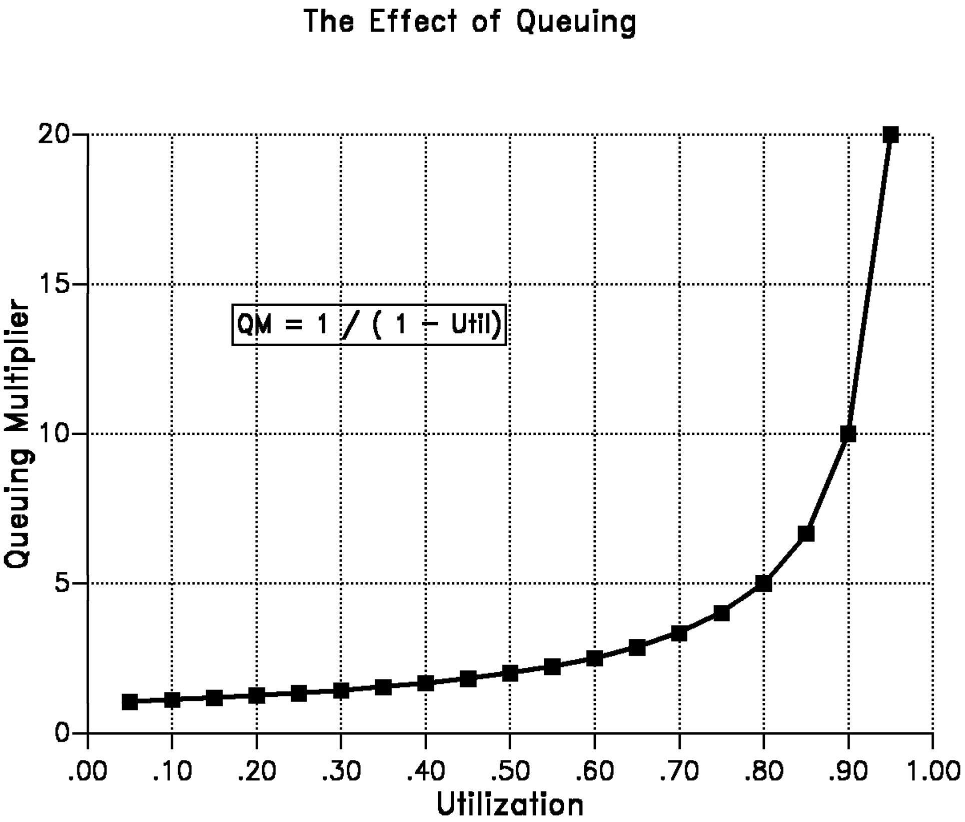


Figure 2. Queuing Effect: Simple Equation for Single Server and Normally Distributed Random

3.3 Additional Queuing Considerations

There are a few additional considerations to be aware of when considering how queueing impacts performance. In particular, this section looks at the effects of multiple servers.

3.3.1 Multiple Servers

The simple queuing theory equation discussed before assumes a single queue of requestors and a single server. In the high-end of the AS/400 product range, some models have multiple processors (N-way) that have more than one central processor executing instructions even though there is only a single queue of CPU requestors (Task Dispatch Queue). In this situation, the increased number of servers reduces the Queuing Multiplier and the average queue length.

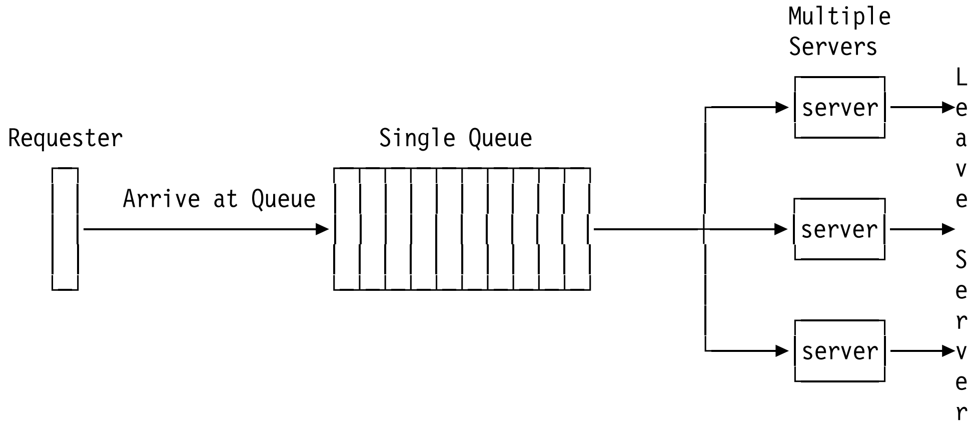


Figure 3. Single Queue with Multiple Servers

Under these conditions, the Queuing Multiplier equation can be represented by a more specific form of the general ″1/(1-U)″ equation shown earlier:

```
        QM = 1 / (1 - U**N)
Where N = number of servers (processors)
      U = utilization
    **N = to the Nth power
```

The following graph highlights the effect of multiple servers on QM:

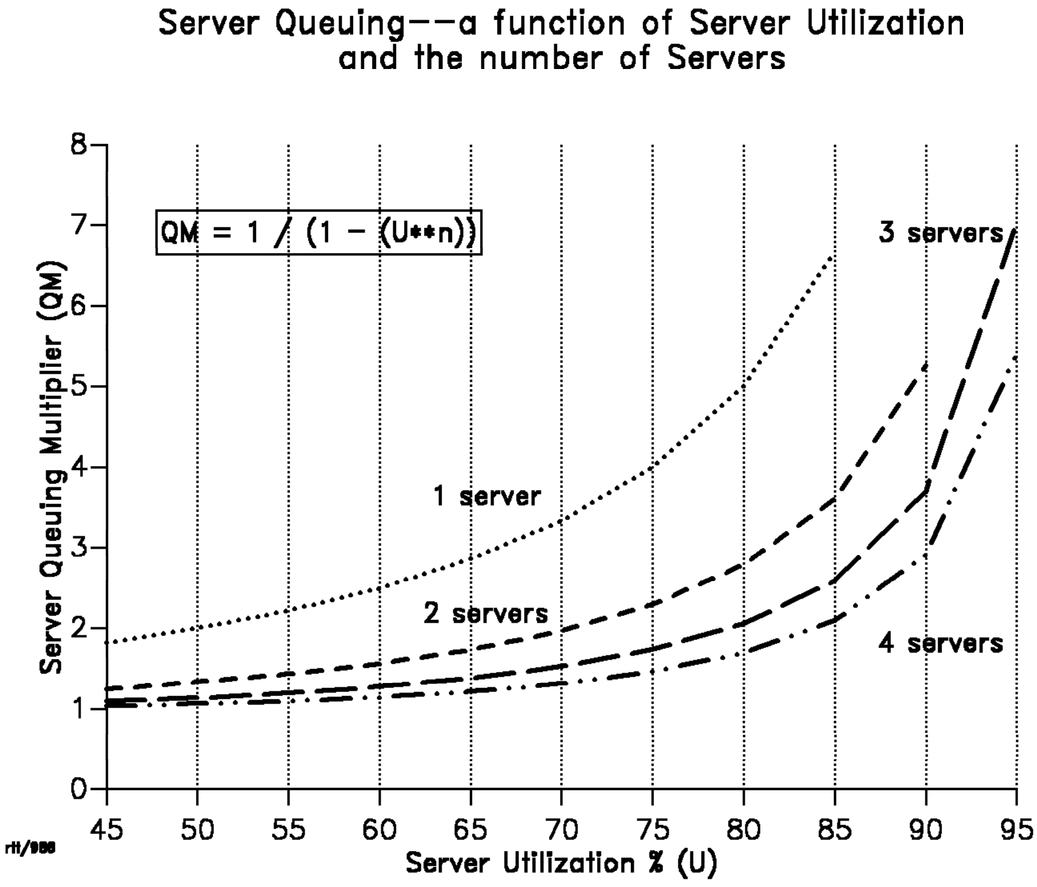


Figure 4. Queuing Effect: Simple Queuing Equation for a Multiple Server

As the chart shows, for a given Queuing Multiplier (QM), the QM value is less (value at that point in the QM versus Utilization graph) is lower as the number of servers increase. This means that for a given CPU utilization value, multiple server systems are less sensitive to increases in utilization than single server systems. The result of this is that you can operate the N-way CPU′s at higher utilizations and maintain more stable response times.

The objective of setting and operating at or below the maximum resource utilization threshold guidelines is to minimize queuing and provide a stable environment for consistent response time.

Refer to Table 33 on page 383 in Appendix A, “Guidelines for Interpreting Performance Data” on N-way processor systems.

3.3.2 Other Factors That Affect Queuing

Two additional factors play an important part in calculating the Queuing Multiplier and predicting it′s effect. They are:

- Server utilization at equal and higher priority
- Number of requestors at equal priority

Use the following equation for more accurate CPU Queuing Multiplier computation:

```
                  1
CPU QM = ------------------------------
         1 - {(U1 + (U2 * (n-1)/n))}**P

Where U1 = CPU Utilization for all higher priority jobs
Where U2 = CPU Utilization for all equal priority jobs
      n  = Number of competing jobs of equal priority
      P  = Number of Processors
```

This equation is the most detailed one used here for CPU Queuing Multiplier calculations. As you can see, the simplified version 1/(1-U) can be derived from this equation.

3.4 AS/400 Layered Machine Architecture

Figure 5 on page 19 illustrates the **AS/400 layered machine architecture**. A primary characteristic of this architecture is the high-level machine interface (MI), which separates Figure 5 into two parts. All licensed programs (LPs) such as OS/400, Query, or OfficeVision/400, and non-IBM application program products (APPs) use an instruction set defined by MI instructions.

The OS/400 portion of Figure 5 represents all of the OS/400 modules. Those illustrated are only a few of the basic supervisory and support functions of the operating system that supports the high-level languages (HLLs).

All HLLs make use of the same OS/400 modules. For example, an open of a database file in C, RPG, COBOL, or CL requires the same set of OS/400 modules. However, all executable programs are bound by the functions available within the programming language.

The machine interface (MI) portion of Figure 5 on page 19 is composed of a set of defined objects and instructions. MI is supported by a programming level called Licensed Internal Code (LIC), which is separated into two distinct classes of support.

One class is the system support, which includes such functions as storage management, database management, and input/output (I/O) support. These can be either **tasks** or **modules**, and are illustrated in further detail in Figure 6 on page 21.

The second class is the translator, or encapsulator, which converts machine instructions into AS/400 RISC instructions. This is essentially a compiler step. At this point, many MI instructions are mapped directly to RISC instructions. Other MI instructions may call a SLIC module that is implemented in multiple RISC instructions. An example of this kind of module is #CCVCN that converts characters to numeric or numerics to characters.

The RISC layer consists of two distinct classes of support. One class pertains to operating system support (as with SLIC). The second class consists of machine instructions and extended function RISC instructions that are executed by the hardware.

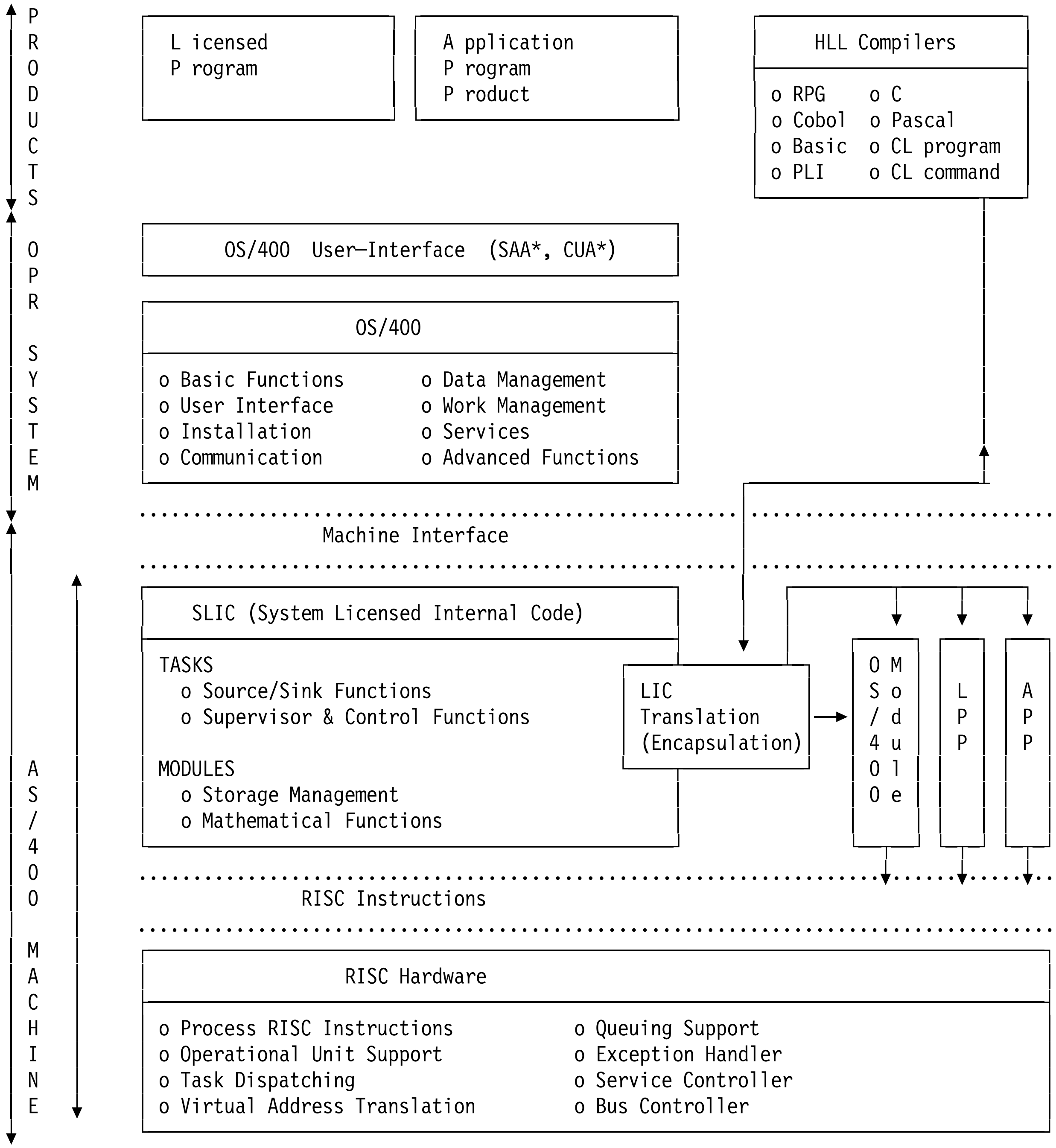


Figure 5. AS/400 Architecture

Refer to the bottom of Figure 6 on page 21 as you read the following sections.

3.4.1 System Licensed Internal Code (SLIC)

SLIC tasks have no job attributes and are for the most part, started at IPL time. SLIC tasks asynchronously perform work requested by both OS/400 modules and SLIC modules. This is in contrast to SLIC modules that are called from a higher level HLL or OS/400 program; they are part of the job that calls them and, therefore, run synchronously within the job.

As you can see from the flow of Figure 6 on page 21, SLIC tasks run below the machine interface and are primarily concerned with I/O operations (whether communications, disk, or device), and usually have a higher priority than a process.

Note: To see which SLIC tasks are running on your AS/400 system, review the *Performance Tools Component Report* or *Transaction Summary Report* when run with the ***HV** option. Each of these reports includes a list of all jobs and tasks that run during the collection period.

3.4.2 Machine Interface—Supervisor Linkage (MI-SVL)

MI-SVLs are illustrated just below the machine interface in Figure 6 on page 21. As the name implies, MI-SVLs are the **link** between HLL or OS/400 programs and the SLIC modules within a calling stack. When performing certain complex MI instructions, HLL programs call MI-SVLs (Supervisor Link Instructions) that invoke the appropriate SLIC modules to perform the work. For more information on MI-SVLs, refer to *AS/400 Licensed Internal Code Diagnostic Aids Version 3*, LY44-4900 and LY44-4901.

3.4.3 OS/400 Modules

OS/400 modules are operating system functions available to all HLL programs. As previously explained for MI-SVLs and tasks, we know that OS/400 modules use the services of existing SLIC tasks and MI-SVLs. For example, QDBPUT (an OS/400 module) writes records to a database file. When more space is required, it calls the services of the storage manager (a SLIC task) to receive the space needed.

Note: The names of all OS/400 modules begin with the letter Q, and have the second and third letter positions representing the module component type. For information on module component types, refer to *OS/400 Diagnostic Aids Version 3*, LY44-4900.

3.4.4 Process

There are many ways in which processes (used synonymously with jobs) are started on the OS/400 V3R6 system. For example, when a user signs on, an interactive process is started, or when a job is started from a job queue, a process is started. These processes are displayed by the WRKACTJOB command.

Some of the processes started by OS/400 at IPL time include SCPF, QSYSARB, and QLUS. Each process has many attributes that are displayed with the WRKJOB command. Within the process, the user programs are called beginning with the initial program shown in Figure 6 on page 21.

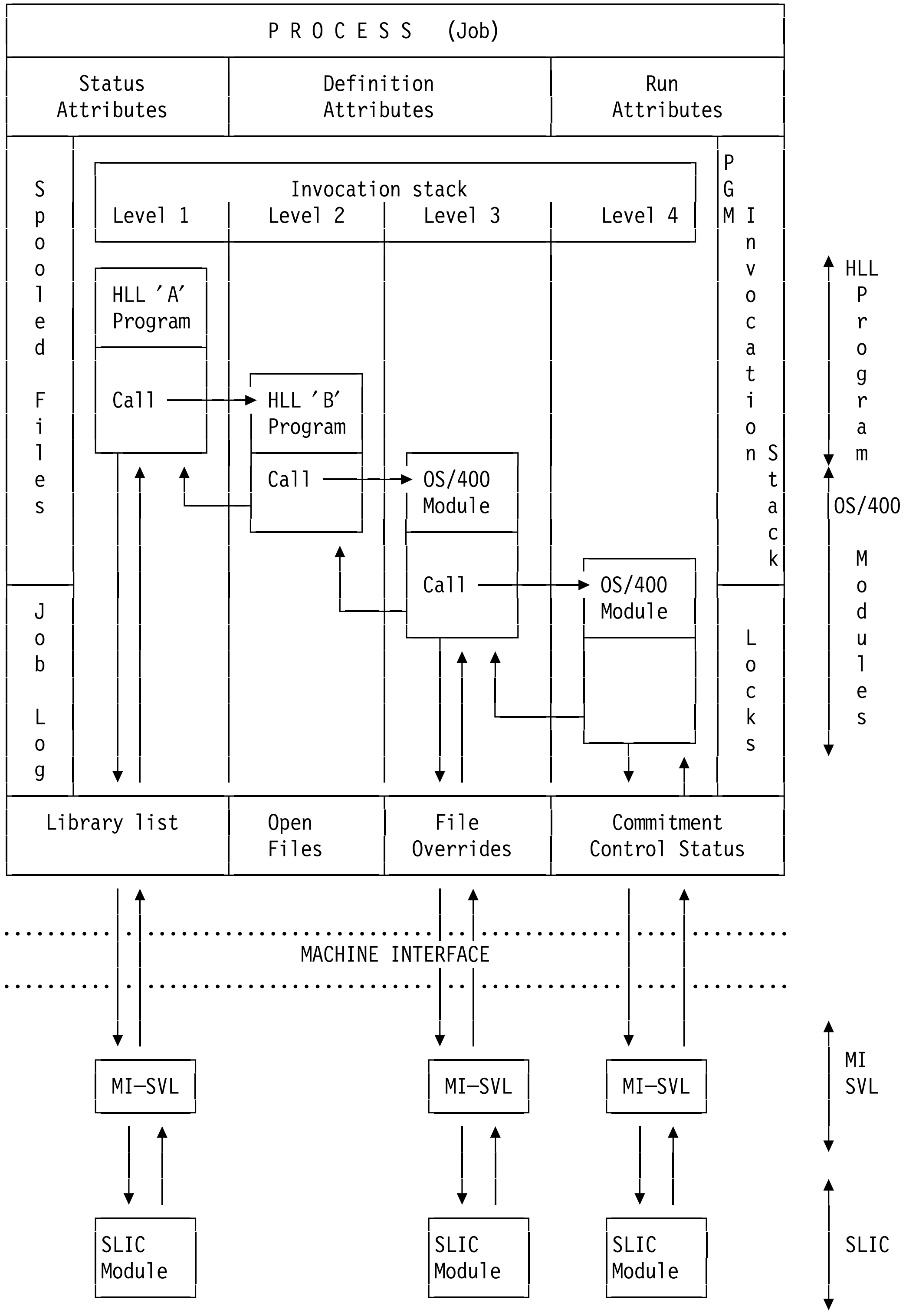


Figure 6. Process, Tasks, OS/400 Modules, and MI-SVLs

Each call to another program creates a new calling stack entry. The creation of a stack entry uses system resource to:

- Open files (if they are not shared)

- Handle overrides
- Allocate Storage
- Initialize program variables (process static storage area)

Normally, the deeper the level of the calling stack, the greater the number of calls on that level. For example, in Figure 6 on page 21, it is not unlikely that program ″A″ is called once, which calls program ″B″ a couple of times, which in turn calls the OS/400 module in level three several times.

In this way, HLL programs repeatedly use OS/400 modules to perform specific work functions, such as opening files or reading records.

Additionally, each HLL program and OS/400 module in the calling stack executes many MI-SVL transitions to SLIC modules. For example, the HLL OPEN statement involves calls to several OS/400 modules that invoke many MI-SVLs. As a result, the work necessary to perform an OPEN is spread across many system components. PEX STATS can be used to determine the distribution of the resources used by each module and the cumulative resources used by a user program that issues an OPEN.

3.5 CPU

The amount of CPU time used by a transaction or batch job is dependent upon many factors:

- Programming algorithms and the structure of data
- Paging and other activity causing additional disk I/O operations
- High system load causing jobs to wait so long for the CPU that they loose some of their main storage pages and require re-paging (thrashing)

Demand placed on the CPU varies depending upon workload, program implementation, and error recovery.

The CPU time is also distributed among different functional System Licensed Internal Code (SLIC) tasks (for example, storage management page-out tasks, asynchronous disk I/O tasks, communications and workstation tasks), some of which are identified in Chapter 26 of the *AS/400 Diagnostics Aids Version 3*, LY44-4900 (available to IBM-licensed customers only).

On many AS/400 systems, SLIC tasks may normally use up to 6% to 12% of the used CPU time, while OS/400 subsystem and monitor jobs normally use about 1% to 3%.

High CPU time in SLIC tasks is often associated with higher than average disk I/O rates (more than one per second) or error recovery. Do not include the storage management tasks (the ones whose name starts with SMAnnn or SMPnnn in this guideline; they have high I/O rates due to the nature of the work they perform. The disk I/O performed by the storage management tasks is counted in the jobs that cause the I/O.

All CPU time used in the system is assigned to the using task, whether it is an OS/400 job or a SLIC task.

3.5.1 CPU Service Time Equation

CPU requirements discussed in this section assume job priority values of 20 or lower (high-priority jobs). This should include all interactive jobs.

The CPU Queuing Multiplier formulas in this section do not take into consideration the delay cost scheduling algorithms used in V3R6. For purposes of this discussion, assume that the jobs run at their assigned RUNPTY value throughout their lifetime.

The average time used by the CPU to service a transaction can be estimated using the following equation:

```
CS = QM * CT                  ; CPU service time (sec/trans)
QM = 1 / (1 - U)              ; Affect of queuing

   CT = CPU seconds per transaction
   U = UTIL * 0.01
   UTIL = CPU utilization for system, spool, interactive tasks
          and priority 20 or higher batch jobs, measured in percent.
```

Note: This formula does not take into account N-way systems. You can modify it using the formulas for the N-way processor shown in Chapter 3, “Factors Affecting Performance” on page 11

Example

Assuming CPU% = 50% and a transaction takes 0.5 seconds of CPU time, the time to use the CPU is about 1.0 second.

```
Assume U = 0.50 and CT = 0.5 seconds

QM = 1 / (1 - U)
QM = 1 / (1 - 0.50)
QM = 2
CS = 2 * 0.50 = 1.0 sec
```

3.5.2 CPU Guidelines

More stable CPU service and queuing time can be expected for interactive work if the CPU utilization for all work at that priority and higher (lower numerical value, for example, if priority 20 work is the current priority, all work at priorities 00-19 are also included) is less than:

- 70% for a single processor
- 76% for a 2-way processor
- 79% for a 3-way processor
- 81% for a 4-way processor

However, close observation is recommended when CPU utilization approaches these levels. For example, for a one-way processor at 70% utilization, an increase in utilization by 5% may result in unacceptable response time. However, for a three-way processor at the same utilization, a 5% increase in utilization may result in acceptable response time.

3.5.3 CPU Relative Performance Rating (RPR)

The AS/400 family has a tradition of rating the relative processor speed among models with a **relative performance rating (RPR)** performance value, where a value of ″1″ represents the original and smallest AS/400 system, the model B10. The RPR′s for the various AS/400 series of systems (9401, 9402, 9404, 9406) are documented in several places including Appendix A, “Guidelines for Interpreting Performance Data.”

For the AS/400 traditional system models, the RPR value represents testing results with an interactive internal IBM benchmark (RAMP-C). This comparative rating is used in rating the AS/400 Advanced Series using PowerPC technology and has proven a reliable guideline for capacity planning of system resources. For the AS/400 Advanced Server model systems, a different internal IBM benchmark involving client/server access methods is used to establish the non-interactive RPR values.

An RPR may not apply to all application environments as their mixture of machine instructions run by the system is significantly different from those generated by the RAMP-C programs and the internal IBM batch and client/server benchmarks.

In addition, consider the difference between an interactive application and a batch application. For example. an F35 has an RPR of 4.7 and an F50 has an RPR of 9.5, roughly two times the F35.

In an interactive environment, the F50 should deliver generally two times the number of transactions per hour that the F35 does. For batch type work, though, you must remember that batch jobs are normally set to less priority (higher job RUNPTY value) than interactive jobs. When no higher priority work is active, the batch jobs should achieve the expected run time when comparing them on the F35 and the F50.

Remember also that the RPR ratings for the multiprocessor models presume that multiple jobs are active concurrently. Run-time expectations for a single, stand-alone batch job should be based on the single processor RPR values of the N-way. The single processor RPR (or relative speed) is the total system RPR value divided by the number of processors. For example, a 320-2052 is a four-way processor rated at 71.5. It′s single processor speed is 71.5/4 or 17.975. If you are calculating how much CPU a job uses and comparing it to what it uses on a 320-2050 (rated at 25.7), the 2050 CPU is 25.7/17.975 or 1.43 times faster than the 2052. Turning it around, the job uses 1/1.43 or about 30% less CPU time on the 2050 than it does on the 2052 for the same amount of work.

One of the most significant reasons for the improved overall performance of the N-way processors is the high reduction in CPU queuing time (the QM) compared to the single processor CPU′s running at the same utilization. This is shown in the curves in Figure 4 on page 16.

3.6 Commercial Processing Workload (CPW)

For all future releases, the Relative Performance Ratings (RPRs) are replaced by CPW ratings. RAMP-C has been used in the past, but with changes in customer environments and dramatic growth both in the raw performance and system capacity of the AS/400, RAMP-C is no longer a feasible workload.

The RAMP-C workload executes significantly fewer instructions per transactions and its memory per user requirements is significantly smaller. The RAMP-C database is relatively small and when used with larger model AS/400 systems, the database resides in memory. This results in minimal I/O and the relative RAMP-C performance becomes solely a function of CPU performance.

CPW is a more sophisticated transaction processing workload. This workload is more characteristic of sophisticated commercial applications particularly on larger AS/400 models.

Note: Although CPW ratings are based on performance workloads similar to TPC-C, CPW cannot be used to estimate TPC-C results or make comparisons reporting TPC-C results.

Table 1 shows a comparison between RAMP-C features and CPW features.

Table 1. Differences Between RAMP-C and CPW

RAMP-C	CPW
Application Description	
Four user-type, each running a single program	All users float between four distinct applications
No batch component	Includes daytime batch
Reads, updates, inserts	Reads, updates, inserts, deletes block inserts, index changes
Each transaction is similar	Each transaction type is significantly different from others
No journal	Journal and commitment control
Path length about 1/4 of the ″typical″ customer	Path length similar to the ″typical″ customer
Relative path length = 1	Relative path length = 2-3
File I/O Description	
Datafiles accessed = 5	Datafiles accessed = 9
# of logical I/Os = 14	# of logical I/Os = 65.5
Data contention is low	Data contention is high
File Scaling	
# of file is variable	# of file is fixed
# of records per file is fixed	# of records per file is variable
Terminal I/O Description	
# of fields in/out = 17/59*	# of fields in/out = 23/77*
# of characters I/O = 133/511*	# of characters I/O = 88/913*
Note: * = Weighted average of all transactions	

In summary, the following items show the different performance metrics which exist, with CPW being the preferred metric:

- **Relative Performance Rating - Internal Processor:** The speed of the main processing unit relative to the model B10. This number is calculated by measuring the processor instruction rate while running the RAMP-C workload. This has been used to position server models until the introduction of CPW values based on the CPW workload.
- **Relative Performance Rating - System:** The performance of a fully-configured system (main storage and disk drive capacity) relative to the model B10. The number is calculated by determining the number of RAMP-C transactions at 70% CPU utilization. Transactions per hour are calculated at 70% because at approximately this value, queuing for the processing unit causes service times to become variable and often lengthen, causing inconsistent end user response times. On faster AS/400 processing units, 80% or higher usage capacities are possible before the effects of queueing become noticeable.
- **Commercial Processing Workload Value:** The CPW value is the relative system throughput for the CPW ***New-order transaction.*** Since the other four transactions have relatively small frequency requirements and since the New Order Transaction (see the following paragraph) is the primary transaction in the benchmark, the throughput is measured in terms of New-orders only. A valid CPW value is achieved when the correct transaction mix is used and at least 90% of the New Order Transactions complete within five seconds.

New Order Transaction

The New Order Transaction is when a customer order of between five and 15 line items is entered for supply from a number of warehouses. It is a medium-weight read-write transaction that is the foundation of the CPW benchmark. The transaction is executed approximately 43% of the time from each terminal. There is a restriction that 90% of the transactions complete within a five-second end-user response time.

There is NO linear relationship between RPRs and CPW values. Based on a B10 having an RPR of 1.0 and a CPW value of 2.9, one should not be tempted to use this as a conversion from one to the other. Use the tables in Section A.1, “System Capacities for AS/400 PowerPC Advanced Systems” on page 384, especially for larger models, since RAMP-C on larger models tends to be more closely related to RPR internal due to its limitations.

3.7 Dynamic Priority Scheduling

The V3R6 AS/400 system has a new optional job dispatching and scheduling function called Dynamic Priority Scheduling. It is implemented in the SLIC layer and can be turned on or off by the user with a new system value called QDYNPTYSCD.

The benefits of Dynamic Priority Scheduling are:

- No job or set of jobs can monopolize the CPU.
- Low priority jobs, typically used for batch work, have a chance to progress.
- Jobs that use too much resource are penalized by having their priority reduced.
- Jobs response time/throughput behaves much the same as fixed priority scheduling.

This new algorithm replaces the methodology used prior to V3R6 systems. For example, a V3R1 system works as follows:

> On AS/400 systems prior to PowerPC technology, all ready-to-run OS/400 jobs and Licensed Internal Code (LIC) tasks are contained in a list called the Task Dispatching Queue (TDQ) and are assigned to run based on the job′s current priority. On the N-way processor models, each job or task has a cache affinity attribute to keep track of the processor on which the job was most recently active. A job is assigned to the processor for which it has cache affinity, unless that results in a processor remaining idle or an excessive number of higher-priority jobs being skipped. The priority of jobs varies very little such that the re-sequencing for execution only affects jobs of the same priority (RUNPTY). This is referred to as Fixed Priority Scheduling.
>
> V3R1 systems have an algorithm to determine how much CPU a job uses while doing it′s last ″n″ disk I/O operations. If the value falls below a certain limit, the job is temporarily promoted to allow it to do another disk I/O sooner. This temporary increase in run priority is not visible with any OS/400 commands or Performance Tools reports.
>
> V3R1 systems favor a job that performs disk I/O by placing it at the front of other jobs of equal priority when the job completes a disk I/O operation.

In V3R6, the new Dynamic Priority Scheduling (DPS) algorithm changes the current priority of jobs whose assigned priority (RUNPTY) is in the range of 10 to 99 and schedules them to use the CPU based on each job′s ″delay cost″. Job priorities one through nine are not included in the DPS processing. Most SLIC tasks are excluded from the algorithm because they run at a priority higher than that of any user or system job.

A job′s Delay Cost is periodically computed by DPS (the SLIC job scheduling functions) based on how long the job has been waiting on the TDQ as well as it′s current priority. If the job′s delay cost indicates that it has been waiting too long without getting any CPU time, the job priority is adjusted to make it more likely to receive CPU time sooner.

> The processor cache affinity attribute is not used in the new AS/400 PowerPC processors. Thus, on the new N-way multiprocessor systems, a job has equal affinity for all processors and CPU scheduling is based only on delay cost.
>
> There is no evaluation and promotion of a job′s priority as a result of the amount of CPU used over the last **n** disk I/O operations on an AS/400 system using the new PowerPC Technology. However, the system does favor a job that has just completed a disk I/O operation by making it more eligible for the processor than other jobs at the same priority and delay cost.

Use the system value QDYNPTYSCD to select whether or not to activate Dynamic Priority Scheduling. DPS uses this system value to determine the algorithm for scheduling jobs running on the system. The default setting is to use Dynamic Priority Scheduling (a value of ″1″). To use Fixed Priority Scheduling, the system value has to be set to ″0″. The DPS scheme allows the CPU resource to be spread to all jobs in the system and prevent low priority job CPU starvation.

One DPS objective is to prevent long running, batch-type interactive transactions, such as a query, from running at priority 20 for the entire transaction time. Another objective is to ensure that lower priority jobs eventually get some system resource. With fixed priority scheduling, when high priority (20) jobs consume as much as 70% or more of the CPU, the lower priority (50) jobs rarely get much CPU time. In this situation, DPS allocates more of the CPU to the lower priority work at a cost of small degradation in the higher priority work.

3.7.1 Delay Cost Terminology

Delay Cost

Refers to how expensive it is to keep a job in the system. The longer a job spends waiting on the TDQ for the CPU, the larger its delay cost. With DPS active, a high delay cost causes the priority 10 through priority 89 jobs priority to increase over time. The same as the priority value, jobs of higher delay cost are given the CPU ahead of other jobs of lower delay cost.

Waiting Time

Used to determine the delay cost of a job at the time the dispatcher inspects the job. The amount of time the job has been waiting on the TDQ to use the CPU affects the new delay cost value.

Delay Cost Curves

The end-user interface for setting job priorities has not changed. However, internally the priority of a job is mapped to a set of delay cost curves (see Figure 7 on page 30). The delay cost curve is used to determine a job′s delay cost based on how long it has been waiting on the TDQ and it′s RUNPTY. The delay cost is used to determine how much to adjust the job′s priority and possibly reposition the job on the TDQ list.

On a lightly-loaded system, a job′s delay cost normally stays at its initial value. If overall system CPU utilization is not at the limit, jobs do not climb the delay cost curve and, therefore, stay at the same priority relative to all other jobs. As the CPU utilization increases, jobs start to climb their respective delay cost curves; however, this has little, if any, effect on job priorities and job dispatching sequence.

As the workload causes the CPU utilization to approach the 80% to 90% level, some of the jobs on lower (flatter) slope curves (lower priority) that have been waiting a long time for the CPU can begin to overtake jobs on higher slope curves that have been waiting in the TDQ for a shorter period of time. This is when the Dynamic Priority Scheduler benefits overall system throughput as the lower priority jobs are able to use some CPU time.

When the CPU utilization is at 100%, the lower priority jobs can climb quite a way up their respective delay cost curve and overtake (that is, their priority becomes higher) jobs on other curves. This is when the Dynamic Priority Scheduler algorithm is working to provide resources to all jobs.

Reset When a job leaves the wait or ineligible state and enters the active state (that is, W->A or I->A job state transition) and is put on the TDQ, it′s cost is at it′s initial value and it′s priority is set to its original assigned priority. Other situations are:

1. Once a job has used some amount of CPU time and reaches the limit of the delay cost, it′s delay cost is minimal and it is ″reset″ which causes it to ″slide″ down its curve to the start of the curve (that is, it is now at the lowest priority value in that delay cost

curve′s range), which may be lower than the jobs assigned priority. This allows other jobs on the same curve to eventually catch-up and use the CPU.

A job′s initial RUNPTY value determines what delay cost curve it is on. For example, RUNPTY(20) is on the Class II Delay Cost Curve; a RUNPTY(23) job is on the Class III curve. When a job on a Class I or II curve reaches the limit of it′s delay cost, it is reset and dropped down one class. As the job′s delay cost subsequently increases again, it moves back to it′s original class curve.

> For example, the Class II curve covers priority 17 through priority 22. When a job with priority 20 goes active, it is placed on the TDQ at priority 20. After any job that is on either the Class I or Class II curve uses a certain amount of CPU time (predetermined by the system), it is ″reset″ and moved back down the curve to the lowest priority represented by that curve (in this case, 22).

2. Jobs on Class III through Class VI delay cost curves are reset to the lowest value for that class when they use an amount of CPU time equal to the machine′s internal time slice value. This value varies from one model to another; it is in the range of 200 milliseconds to 400 milliseconds with the smaller value being used on the faster models.

3. A job′s delay cost and priority is reset to it′s original assigned value when it goes active (is put back on the TDQ) after a Lock Wait, after a Short Wait Extended, or it reaches time slice end (it does not matter if the state transition is A->A or I->A.

4. A job is not reset when it goes active after a Short Wait or Receives a message from a data queue and does not have to wait (for example, go from Active to Wait and back again) because a message is already on the queue to be received. See the information on page 32 for additional discussion.

Note:

1. Jobs that leave the activity level are reset to their assigned run priority when they return to the activity level.
2. Jobs that have been promoted as a result of seize/lock priority adjust are not reset while they are in the promoted state. They are re-evaluated sometime after the seize/lock has been released.

3.7.2 Priority Mapping to Delay Cost Curves

The mapping scheme divides the 99 ″user″ job priorities into two categories:

- User priority 0 through priority 9:

  This range of priorities is for critical jobs such as system jobs or high priority applications that should not be delayed by other application jobs. Jobs in this range are **not** overtaken by user jobs of lower priorities. This is true even though it is possible that a job′s delay cost can cause it′s priority to be higher than that of a priority 9 or higher (lower number) job. Jobs with an ***assigned*** priority 0 through priority 9 are dispatched before any job that is on any delay cost curve class regardless of the other job′s delay cost.

Note: You should generally not assign long-running, resource intensive jobs within this range of priorities.

- User priority 10 through priority 99:

  This range of priorities is for jobs that execute in the system with dynamic priorities. In other words, the dispatching priorities of jobs in this range can change if their delay cost is high and DPS is active (the QDYNPTYSCD system value is set to ″1″).

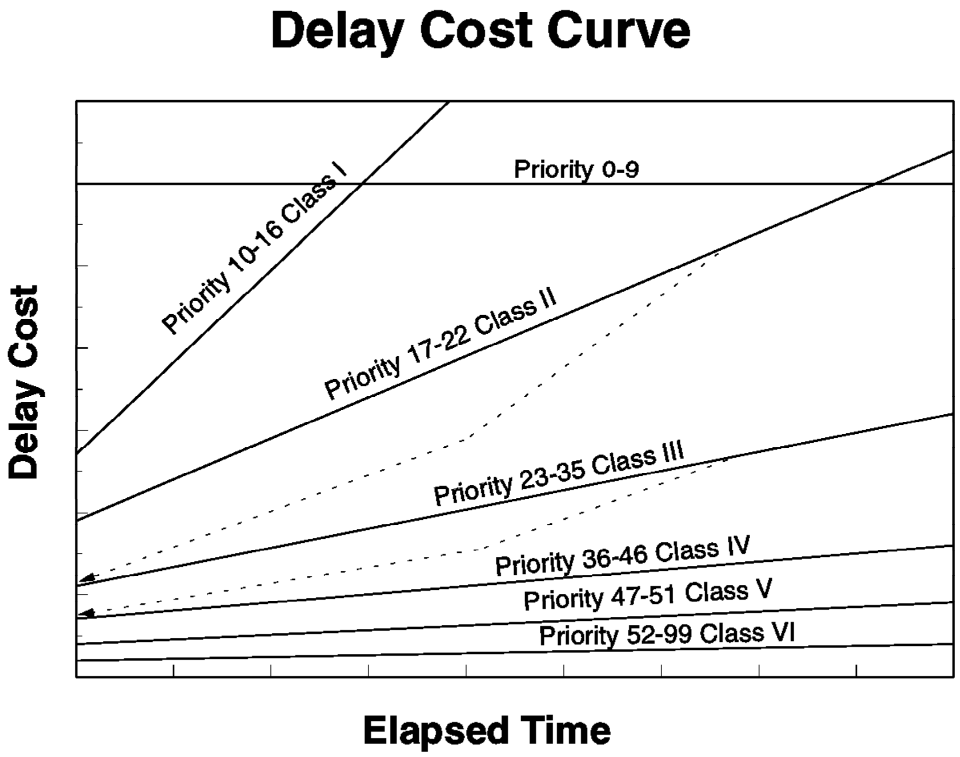


Figure 7. Dynamic Priority Scheduler Cost Curves

The priorities in this range are divided into groups and are assigned to different delay cost curve classes that are defined as follows:

```
Class   I,  Priority 10-16, slope 32, resets only to itself
Class  II,  Priority 17-22, slope 16, resets to Class III
Class III,  Priority 23-35, slope  8, resets to Class IV
Class  IV,  Priority 36-46, slope  4, resets only to itself
Class   V,  Priority 47-51, slope  2, resets only to itself
Class  VI,  Priority 52-99, slope  1, resets only to itself
```

Jobs in the same class have the same CPU resource usage limits. Internally, each class is associated with a unique delay cost curve. This gives some preferential treatment to jobs of higher user priorities at low system utilization similar to the effect of fixed priority scheduling.

The slope of the delay cost curve controls how quickly the delay cost changes and influences the rate of priority changes to jobs in that class. The slope is defined as the quotient of the delay cost divided by the wait time. For example, a job in class I moves further up the cost curve than a job in

Class II or higher moves in the same amount of wait time. The slope values shown in the preceding table are meant to represent relative values. That is, jobs in Class I move up their delay cost curves twice as fast as jobs in Class II, four times faster than Class III, and so on.

With this mapping scheme and using the default priorities of 20 for interactive jobs and 50 for batch jobs, users generally see that the relative performance for interactive jobs is better with DPS with the benefit that batch jobs experience less CPU starvation.

3.7.3 Testing DPS and Results

In this section, the results of two specific measurements demonstrate the effect of the Dynamic Priority Scheduler. It should be noted that the results vary according to the AS/400 workload used, and customers should validate these results using their own workload to measure the effects of DPS.

The values shown in Table 2 are the result of testing an environment consisting of:

- An interactive workload of many simple transactions running at priority 20 and consuming approximately 70% CPU utilization
- A CPU-intensive interactive job running at priority 20

Table 2. Effect of Dynamic Priority Scheduling: Interactive Only

Measurement	**QDYNPTYSCD = ′1′ (ON)**	**QDYNPTYSCD = ′0′ (OFF)**
Total CPU Utilization	93.9%	97.8%
Interactive CPU Utilization	77.6%	82.2%
RAMP-C Transactions Per Hour	60845	56951
RAMP-C Average Response Time	0.32 sec	0.75 sec
Priority 20 CPU Intensive Job CPU	21.9%	28.9%

The values shown in Table 3 on page 32 are the result of testing an environment consisting of:

- An interactive workload of many simple transactions running at priority 20 and consuming approximately 70% CPU utilization
- A CPU-intensive batch job running at priority 50

Table 3. Effect of Dynamic Priority Scheduling: Interactive and Batch

Measurement	**QDYNPTYSCD = ′1′ (ON)**	**QDYNPTYSCD = ′0′ (OFF)**
Total CPU Utilization	89.7%	90.0%
Interactive CPU Utilization	56.3%	57.2%
RAMP-C Transactions Per Hour	61083	61692
RAMP-C Average Response Time	0.30 sec	0.21 sec
Batch Priority 50 Job CPU	15.0%	14.5%
Batch Priority 50 Job Run Time	01:06:52	01:07:40

3.7.4 DPS Conclusions/Recommendations/Tips

- When there are many interactive jobs running on the system and you want to ensure that one CPU intensive interactive job **does not take over** (see Table 2 on page 31), Dynamic Priority Scheduling gives you the desired result. In the test cases, the interactive jobs have higher transaction rates and faster response times, and the priority 20 CPU intensive job received less CPU.

  There are applications in which there is equal priority batch and interactive running concurrently. In many current AS/400 applications, this is often the case in manufacturing, distribution, or banking applications such as ATM or credit card authorization. Often these applications contain jobs that run as batch jobs at priority 19.

  If DPS is used and these types of jobs are in Class II along with the interactive work, it is possible that the higher priority ″batch″ job drops to priority 22 and stays there for a period of time until it′s delay cost raises it back up to priority 20. If this takes a few seconds due to other workload on the system, the response time requirements of the higher priority batch may not be satisfied. In this case, the setup should be as follows:

  - Put the high priority jobs at priority 16 (Class I). Class I jobs do not reset into Class II.

    This prevents a higher priority Class I job from having to compete with the lesser priority Class II interactive jobs. The class I jobs should normally have low CPU requirements and consistent response time requirements. To maintain the response time goals over varying interactive load conditions, you have to ensure that they never have their priority lower than the interactive priority, which can happen if they all share Class II.

- The case when data queue messaging is used may require special consideration if DPS is used. In these applications, a job can be reset to the lowest priority in a class even though it has new work to perform. This can happen when a Receive Message is satisfied because there is a message on the queue at the time of the Receive and DPS does not reset the original cost and priority. Therefore, the job stays at the lowest priority in the class for a period of time and has to wait until it′s delay cost rises to equal or above that of other new work before it can start processing.

The recommendation is that for any application that can have high volume message peaks and shares a DPS class with other jobs, all of the jobs in that class are assigned the highest (numeric) priority in that class. For example:

> An AS/400 V3R1 environment is to be moved to a V3R6 system and has interactive applications at priority 20 and credit card authorization at priority 19 or 20. These jobs should be assigned to the same DPS class and they should all be assigned to an initial priority that is equal to the highest (numeric) value in the class (all Class II at priority 22). This ensures that newly-arrived credit card authorization messages do not have to wait a few seconds before they start processing.
>
> The same logic applies to a manufacturing or distribution picking application where the picking jobs are considered by the business to be equal to or even higher priority than ***normal*** interactive work. Often when using this type of processing, the picking application runs at priority 19 and other interactive applications run at priority 20. If the Dynamic Priority Scheduler is used with the same priorities, there is a high probability that the priority 19 jobs do not behave the same as they do under Fixed Priority Scheduling.

- Dynamic Priority Scheduling ensures that lower priority batch jobs get some of the CPU resources without significantly impacting the interactive jobs (see Table 3 on page 32). In the test case, the interactive workload gets less CPU resource and results in slightly lower transaction rates and slightly longer response times. However, the batch job gets more CPU utilization and, consequently, shorter run time.
- It is recommended that you run with Dynamic Priority Scheduling for optimum distribution of resources and overall system performance.
- It is strongly recommended that you use the Dynamic Priority Scheduling algorithms to ensure system operational integrity.

  If you use Fixed Priority Scheduling, there is an exposure that a single job or set of relatively high priority jobs can monopolize the CPU and cause significant degradation of all other jobs in the system. Prior to V3R6, there was a Licensed Internal Code (LIC) function to detect and correct the problem by temporarily ***suspending*** these ***high CPU consumers*** until all other jobs were given some CPU time. This algorithm allowed a low priority (high number) job that held a system level lock to eventually get enough CPU time to free the lock.

  This function (called the ***hog tracker*** in V3R1 and prior releases) is no longer available in V3R6; therefore (if you are not using DPS), there is a possibility that a high priority, high CPU job can cause a long delay to other jobs in the system and perhaps even necessitate an IPL (if the job cannot be cancelled).

3.8 Task Dispatching Overview

Understanding AS/400 task dispatching is not required for most operating environments. However, a customer with performance concerns must understand the importance of job or system task priority in getting work done. The run priority of a job is assigned by the RUNPTY parameter of the Class Description (system defaults are provided). The lower the numerical value, the higher is the priority of the job. A running job can be preempted by a job or a system task of higher run priority.

The priority of System Licensed Internal Code (SLIC) tasks are predetermined. Some run under a user job priority while others run at high priority (higher than user jobs).

A basic understanding of task dispatching can be helpful when analyzing problems such as performance degradation or understanding that the algorithm on an AS/400 server model modifies normal usage of the job priority (RUNPTY) attribute for task dispatching.

Task dispatching on the AS/400 system is controlled from a single Task Dispatching Queue (TDQ). All OS/400 jobs and SLIC tasks appear as Task Dispatch Entries (TDE) in this queue.

The high-end multiprocessor (N-way) systems also share a single TDQ. Refer to Section 3.3.1, “Multiple Servers” on page 15 for discussion on a service system with a single queue with multiple servers. These TDEs are sorted by priority in the TDQ. The selection of a task for execution is based on a combination of the following values:

- Priority
- Eligibility

Selecting which tasks (OS/400 jobs and SLIC tasks) to run in which processor is done through a combination of priority, eligibility, and for non-PowerPC AS/400 system′s cache affinity. If all processors are busy with higher priority tasks, an eligible task must wait until a processor is free before it is dispatched. In PowerPC processors, all tasks have equal affinity to all processors at all times.

Note that in AS/400 server models, the interactive jobs have a much shorter internal time slice value compared to the time slice value used for non-interactive jobs. This assists the system in favoring non-interactive jobs. However, should several interactive jobs running at a RUNPTY higher (lower value) than non-interactive jobs perform long running batch-like functions, non-interactive job performance may degrade.

3.8.1 AS/400 Multiprocessor Main Storage Interleaving

For the AS/400 Advanced Series that did not use PowerPC Technology (often referred to as CISC processors), there are memory card placement considerations rules that maximize performance especially with multiple processors. It is possible to get a significant performance advantage by filling the first four memory slots with the same size memory cards. This allows ″maximum interleaving″ of storage access requests by the system code. Without interleaving, the first memory card can become heavily utilized in the multiprocessor configuration. With interleaving, there is a potential performance improvement of more than 15% over a similar but non-interleaved system.

For high-end Advanced Series AS/400 models with PowerPC (often called RISC models), on 500 (#2142), 510, and 530 and related server models, there are four memory slots and memory must be installed in pairs.

Due to differences in system design, the reasons for the performance difference in CISC are not applicable to RISC. There is only about a 3% advantage in spreading memory across all four slots versus having the equivalent amount of memory in the first two slots. Also, RISC is insensitive to an imbalance in the relative size of the memory pairs (compare to CISC where the first four cards needed to be the same size for maximum performance).

3.9 Memory

Memory is a key resource contributing to overall response time.

The use of main storage by a job cannot be directly measured with any of the available performance tools, nor is it possible to determine how much memory a particular job is using. On V3R6, there are new guidelines for memory usage, both for the total system and for individual jobs. See Appendix A, “Guidelines for Interpreting Performance Data” on page 383 for details on memory guidelines.

3.9.1 Database and Non-Database Page Faults

The effect of memory demand can be observed, and measured to a certain degree, by using page faulting rates in memory pools. A page fault is a notification that occurs when an address that is not in main storage (memory) is referenced by an active program.

A database (DB) fault occurs when referencing database data or access paths. A non-database (NDB) fault refers to objects other than database objects. NDB pages include programs, data queues, configuration objects, internal space objects, and so on.

When either type of fault occurs, the job must wait for the necessary information to be transferred from disk to main storage. A disk operation that the job must wait for is called synchronous disk I/O.

When a database or non-database page fault occurs, storage management uses a default number of pages to read in. These defaults are:

- Database object:

  A single page of 4096 bytes per page is brought from disk to memory.

- Non-database object:

  The number of pages read in is based on the specific implementation of the object. The default is a single page (4096 bytes).

Other system functions and individual product implementations have their own default values. There is no known list of implementation details in this area, but the following options may be used to override the defaults:

- Database SEQONLY(*YES number-of-records) on the Override Database File (OVRDBF):

  AS/400 database support determines the number of pages to bring in.

- Database MAXRCDS(number-of-records) on the Override Database File (OVRDBF):

  AS/400 database support determines the number of pages to bring in.

- Use of expert cache support:

  AS/400 database support and storage management support determine the number of pages to bring, based on current ″reference pattern″ statistics for objects supported by the expert cache algorithm.

  See this chapter and Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413 for additional information.

- Using the Set Object Access (SETOBJACC) command:

  Use the SETOBJACC command to load a database file, database index, or a program into main storage. The most effective use of this command is when you load the object into a ″private″ pool in which there are no jobs running and, therefore, no page faults occurring. You can load multiple objects into the pool. How many objects can be loaded is a function of the pool size and the object sizes. The command returns a message that tells you how many bytes of the object were loaded and how much storage was in the pool before the object was loaded. This allows you to manage the contents of the private storage pool.

  SETOBJACC should be used to load objects only into storage pools that are not having their size automatically changed by a storage pool tuner such as QPFRADJ. Because there is no paging in the pool that the objects are loaded into, the QPFRADJ tuner eventually removes most of the pages from the pool to satisfy demand in other pools.

  Use the SETOBJACC command′s PURGE option to remove the object from main storage. This option removes all of an object′s pages from all main storage pools and writes the changed pages to disk.

  See this chapter and Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413 for additional information.

- Object header ″block transfer″ of up to eight pages:

  The implementer of a particular object, such as an ILE C/400 object or an AS/400 data queue, determines if system defaults are used or a block transfer of up to eight pages is brought in.

  AS/400 storage management recognizes this information in the object header.

3.9.2 Memory (Storage) Pools

Main storage is segmented into memory (storage) pools to help manage storage contention. A page fault shows that the referenced page is not in memory (main storage), and is a measure of this contention.

Main storage in use by a job depends, at a specific time, on the job′s size and the demand for memory made by the job and other concurrent jobs sharing the same main storage pool. The main storage requirement is affected by:

- The program size
- Whether or not the program used by the job is shared by other jobs
- The amount of temporary storage in use (such as file buffers and program variables)

Some memory is used for job associated temporary data and is held in an object called a Process Access Group (PAG).

Storage for some system functions, such as that used for the display file RSTDSP(*YES) and DDS WINDOWS functions, occupies space in the job′s storage pool.

Pool paging rates shown in the WRKSYSSTS command displays and the Performance Tools System and Component reports are generally used to infer demand for main storage.

Note: The WRKSYSSTS pages per second rate includes pages read into the storage pool, not just pages brought in due to a page fault. Sometimes the *number of pages rates* and *fault rates* are used interchangeably. However, in some cases, pages are read into the storage pool ″ahead of use″ such as when SEQONLY(*YES) or NBRRCDS(nnn) is specified for a database file. These anticipatory page reads are not recorded as page faults. When a page fault occurs, and DB or NDB pages have to be read in, the job needs additional main memory. This might require that a page frame be ″stolen″, possibly from some other job in the same storage pool. The storage management functions inspect the storage pool for an available page (or pages, depending upon the request). Changed pages in memory are not stolen until they have been marked as written to the disk. The page stealing algorithm selects the least recently used page within each pool. Pages to be stolen are prioritized from most likely to least likely in the following order:

1. Database pages
2. Job (or process) specific temporary pages (usually PAG pages such as file control blocks)
3. Shared, permanent NDB pages, such as programs, message queues, and so on.

If changed pages are detected during the page selection process, they are given to the system page out tasks (SMPOnnn) for writing to disk and are ignored during this pass through the page stealing selection.

One conclusion you can draw is that insufficient main memory shared by multiple jobs can cause increased CPU and disk usage. This is unproductive and leads to diminished throughput and response time. This is especially true for batch processing. Normally both the database and non-database page fault rates for batch pools should be quite low. If there is a lot of non-database object processing such as data queues, user spaces, or other objects that contain user data, you might see higher non-database object faults and page rates.

If you see high database faults in a batch pool, it might mean that there are too many jobs competing for the pools memory and everyone is slowing down. You can try splitting the batch into separate pools or reducing the number of concurrent batch jobs in the pool by reducing the MAXACT value for the subsystem or possibly reducing the activity level of that pool. Do not over-commit the system by running more jobs than the CPU can handle. For example, if each of your batch jobs averages 15% of the CPU in a stand-alone mode, there is nothing to be gained by allowing more than six of them (90% of the CPU) to run at once. Remember that the system overhead is probably in the 6-10% range. It causes additional CPU overhead to move them in and out of the activity level and may also cause unnecessary stealing of one job′s pages by another job that was just let back into the activity level.

Note: A user can assist OS/400 in managing storage pools by assigning jobs of similar CPU utilization characteristics and priorities to the same storage pool. For example, assigning a high CPU utilization job with a high run priority compared to other jobs in the same storage pool usually has a significant degradation impact on these other jobs.

Jobs are assigned to storage pools through the routing entries for a specific subsystem. The Add Routing Entry (ADDRTGE) command and Change Routing Entry (CHGRTGE) command include compare values that can be used to route different jobs into different storage pools. Knowledge of the routing entries used for client/server applications such as Client Access/400, LANRES/400, and ADSM/400 can assist in managing CPU and memory resource utilizations of these IBM products along with customer applications. Refer to Chapter 9, “Performance Problem Analysis” for additional work management information on these client/server applications.

3.9.3 SETOBJACC

OS/400 V2R2 introduced the capability to load specific objects into a specific shared or private storage pool.

The Set Object Access (SETOBJACC) command can load a database file, a database index, or a program into a storage pool. This causes a change in the normal main storage processing for the named object and specified storage pool. SETOBJACC allows the user to exercise some control over the contents of a storage pool. The storage pool can be a private pool, the pool the job is running in, or one of the system′s shared pools.

If a private pool contains only the preloaded ″data″, the data stays in memory until the object is explicitly purged (using SETOBJACC), overlaid with another file as a result of another SETOBJACC (not recommended), or the pool is cleared (using the Clear Pool (CLRPOOL) command).

If possible, the CLRPOOL and SETOBJACC process should be performed when the system work has generally been quiesced. CLRPOOL ensures the pool is cleared of all objects and SETOBJACC loads the designated object into the assigned storage pool.

SETOBJACC allows changes in AS/400 application design methodology, primarily because of the major improvements during random processing of database files. If the application currently does GET by key or GET by key followed by GET next, the data is read synchronously one record at a time. With SETOBJACC, if the file fits and there are no jobs active in the pool, the system loads it into the pool and does not purge the data thereafter. The general recommendation is to load objects only into private pools in which there are no jobs running. This provides efficient application processing regardless of how the file is accessed by the application. Since much of the I/O for random processing is eliminated, it can eliminate the need to sort or reorganize the data for files that fit in a pool.

SETOBJACC support should be used when the key database object (or objects) or program object (or objects) of an application are known, the size of the object (or objects) are known, and a sufficient amount of main storage is available to contain the **entire** object.

See Appendix E, “OS/400 Expert Cache and Set Object Access Overview” for more information on SETOBJACC support.

3.9.4 Expert Cache

Expert cache enhances the AS/400 single-level storage use of main memory as a cache. The same as SETOBACC, it overrides normal storage management algorithms discussed in Section 3.9.2, “Memory (Storage) Pools” on page 36. Expert cache causes frequently accessed data to remain in storage for long periods of time. However, not the same as SETOBJACC which requires the user to know which objects should be specifically placed into storage, expert cache gives System Storage Management the responsibility for *determining and* *managing portions of objects (both database and non-database objects)* that should remain in main memory longer than under normal page management algorithms.

Expert cache is enabled by the user for a *shared storage pool only* and applies to all objects in that storage pool. Expert cache monitors the application environment changes over time and manages how long the objects or portions of an object stay in main storage.

See Appendix E, "OS/400 Expert Cache and Set Object Access Overview" for more information on expert cache support.

3.9.5 Working Set Size

Working Set Size is defined as the amount of main storage necessary to run a program within a job and achieve the best performance. The concept is straightforward, but is not actually measured by any performance tool. It can only be **inferred**.

The recommended way to infer working set size is to start with a small user-defined storage pool and run only a single job of the appropriate application in that pool. Use a procedure in which you make multiple runs as follows:

1. Clear the pool (CLRPOOL to get rid of any used pages so they do not just get moved to the *BASE pool).
2. Change the storage pool size.
3. Rerun the test until page faulting is well within the paging guidelines and the job performance is acceptable.

You may want to repeat this procedure using more jobs each time to see what the optimum storage pool size setting should be.

In V3R6, virtually all of the microcode was rewritten (now called System Licensed Internal Code (SLIC)). The new AS/400 PowerPC technology is 64-bit (versus 48-bit on non-PowerPC AS/400 models), so the nature of this implementation consumes more main storage. Entities such as pointers (used by programs to reference locations in memory) are all 16-bits longer. Also, the default size of a disk page that is brought into main storage has increased from 512 bytes to 4096 bytes, resulting in performance improvements but has also increased the use of main storage. Additional hardware is now supported such as the Integrated PC Files Server (previously known as the File Server I/O Processor or FSIOP), Wireless LAN, and new tape devices. TCP/IP, Client Access/400 host servers, Integrated File System, and SPEC 1170 Openness APIs have been integrated into OS/400.

Because of these changes, additional main storage is needed to achieve the same performance level as a previous release. This is considered as the *working set size* of the system.

To estimate the initial size of storage pools for V3R6, use the guidelines and tables in Chapter 14, Performance Tuning of *OS/400 Work Management Version 3*, SC41-4306.

The storage pool size recommendations for AS/400 PowerPC systems have changed significantly from the previous AS/400 models and must be reviewed and changed before running anything else on V3R6.

Once you have installed V3R6, you should immediately tune your storage pools to ensure proper storage pool allocation for maximum performance. This can be done manually using the WRKSYSSTS command, or automatically using the QPFRADJ system value and associated QPFRADJ system job support.

If using WRKSYSSTS support, observe the page faulting rates in the machine pool during the appropriate system workload and change the pool size according to the ″good″ performance guidelines discussed in Section 3.9.6, "Memory Guidelines" on page 41.

If using the QPFRADJ system value support, the following approach is suggested:

1. Set the QFPRADJ system value to dynamic, run-time tuning (CHGSYSVAL QPFRADJ(3)).
2. Allow the tuning to take place over a two-hour period during peak system activity. This should cause the proper adjustments to pool sizes.
3. If you normally run with QPFRADJ set to 0 (no adjustment), do CHGSYSVAL QPFRADJ(0) after you observe that performance is equivalent to or better than the previous release.

Note: This tuning is recommended for all systems, regardless of their previous release page fault rates or CPU utilizations.

However, any pre-V3R6 system that is approaching the following ″poor performance indicators″ should be carefully analyzed to correct it′s current critical workload environment **before installing V3R6** and perform capacity planning analysis if significant additional work is considered after V3R6 installation:

- Page fault rates that approach the ″poor performance range″ as documented in Appendix A, "Guidelines for Interpreting Performance Data" on page 383 in Table 34, Table 35 on page 383, and Table 36 on page 384
- Interactive application environment with CPU utilization above 70% for interactive and higher priority work
- Batch application environment with over 90% for all batch and higher priority CPU utilization **and** batch job run time already close to the limit of a ″run-time window″

Note: These are general recommendations for any new release installation. It is necessary to set the memory sizes according to the guidelines for all V3R6 installations, especially for systems with small main storage sizes.

If you anticipate additional system workloads after V3R6 installation, we recommend that you use the Performance Tools/400 BEST/1 support for capacity planning **before installing V3R6**.

Note: If the current release is ″on the edge″ of poor performance, and the faults rates are exceeding guidelines, one possible way to correct it is to install additional main storage. If you have considerable workload growth plans, BEST/1 may recommend the addition of more resources. See Section 3.9.6, "Memory Guidelines" on page 41 for more information on page fault guidelines.

3.9.6 Memory Guidelines

Paging activity in storage pools can be caused by either OS/400 jobs, IBM-provided application jobs, user application jobs, or Licensed Internal Code (LIC) tasks. Generally, the range of 15 non-database faults per second or less in the machine pool is considered as acceptable.

There are new guidelines for V3R6. These new guidelines are listed in the *OS/400 Work Management Version 3*, SC41-4306 and Appendix A, "Guidelines for Interpreting Performance Data" on page 383. There are considerable differences between V3R6 guidelines and those for earlier releases of OS/400.

Acceptable Performance and Page Fault Guidelines

On systems that are performing well, high page fault rates should not cause unnecessary concern if the system application environment is relatively constant. While it is always good to understand why page fault rates are higher than guideline rates, the user should not be overly concerned with high fault rates in these cases.

If you are experiencing poor performance, look at page faulting as a possible cause of the poor performance. If additional work is planned to be added to the system, such as 10 more workstations or a new application, high page fault rates should **always** result in using BEST/1 capacity planning to model the increased workload impact.

3.10 Main Storage Sizing Guidelines

Larger amounts of main storage are required to take full advantage of the performance of the new AS/400 Advanced Series using PowerPC technology. The new AS/400 models are provided with substantially more main storage in their base configurations.

The increase in main storage requirements is basically due to two reasons:

- PowerPC RISC architecture, requires more simple instructions to execute the same program as on CISC which uses fewer complex instruction. This does not mean that a function takes longer to execute, but it does result in a function requiring more main storage. This obviously has more of an impact on smaller systems where fewer users are sharing the program.
- The main storage page size has increased from 512 bytes to 4096 bytes (4KB). The 4KB page size is needed to improve the efficiency of main storage management algorithms as main storage sizes increase dramatically. For example, 4GB of main storage is available on AS/400 Advanced System model 530.

  The impact of the 4KB page size on main storage utilization varies by workload. The impact of the 4KB page size is dependent on the way data is processed. If data is being processed sequentially, the 4KB page size has little impact on main storage utilization. However, if you are processing data randomly, the 4KB page size probably increases the main storage utilization.

The minimum memory available on AS/400 PowerPC systems is 32MB versus 8MB on CISC systems. In most instances, 8MB CISC systems require 32MB if you run the same workload when upgrading. However, if the 8MB CISC system is over-committed in main storage utilization, 64MB of main storage may be required when using the AS/400 PowerPC technology.

As a first approximation of the main storage required when moving to AS/400 models using PowerPC AS technology, use the following guidelines. If you are adding additional work as you upgrade to the new models, you should first determine what main storage is required on CISC for this new workload before using the following guidelines.

Table 4. Main Storage Size Guidelines

Main Storage Size on CISC	**Main Storage Size on RISC**
Up to 160MB	(2 X CISC Main Storage) + 16MB (See Note)
Greater than 160MB	2 X CISC Main Storage

Note:

The 16MB that is added is primarily due to the increase in size of the operating system code that must be resident in main storage. It is important to take this increase into account when sizing systems with lower amounts of main storage.

This initial estimate should be followed up by a more detailed analysis with BEST/1 for OS/400. Using BEST/1 to analyze your workload takes into account how main storage is being utilized on your current system.

Refer to Section 12.12.5, “Run-Time Working Set Size” on page 295 and Section 12.12.6, “Compiler Working Set Size” on page 295 for main storage sizing considerations for application development environments.

3.11 Memory Tuning

The Performance Adjustment support (QPFRADJ system value) that is used for initially sizing memory pools and managing them dynamically at run time has been enhanced to support the new AS/400 PowerPC hardware and V3R6.

The following changes were made for tuning done at IPL time (the system value QPFRADJ is set to 1 or 2):

1. The calculation for the minimum machine pool size has been updated to reflect changes in the amount of storage needed for lines, controllers, and devices. The algorithm also has been changed to use (as a base value) the minimum machine pool size calculated by the System License Internal Code (SLIC) instead of the tabular method (based on main storage size) that was used in previous releases.

2. The pool size calculation for the *INTERACT and *BASE pools has been updated. After the machine pool and spooled pool sizes have been determined, 70% of the remaining storage is given to *INTERACT. The remaining 30% is given to the *BASE pool. To better support the unique demands of the client/server environment, on server models, the amounts are the opposite (70% to *BASE, 30% to *INTERACT).

The following changes were made to dynamic tuning done at run time (the system value QPFRADJ is set to 2 or 3):

1. With V3R6 and V3R7, the page fault guidelines stated in the *OS/400 Work Management Version 3*, SC41-4306, are significantly higher than they are for V3R1. The increases are due to the inclusion of additional index faults in the counts for V3R6 and V3R7, and other objects such as faults on storage management directories, user profiles, libraries, and file access paths.
2. The Dynamic Tuner no longer uses the guidelines published in the *OS/400 Work Management Version 3*, SC41-4306, for user pools (any pool except the machine pool). The Dynamic Tuner now calculates a run-time guideline based on the number of active jobs in the storage pool. The type of pool (*INTERACT, *SPOOL, or other shared pool) is also taken into consideration. The run-time guidelines are much different than the published guidelines, especially if the number of jobs is much lower or much higher than the number of jobs for which the system is rated.
3. For Advanced Server models, the *BASE pool is treated with higher priority than the *INTERACT pool. This means that if the Dynamic Tuner determines that both *BASE and *INTERACT require more memory, *BASE gets more memory before *INTERACT. On traditional models, *INTERACT gets higher priority.
4. The minimum pool size for an active pool has been increased.
    - *INTERACT - 3000K
    - *SPOOL - 256K
    - *SHRPOOL1-10 - 1000K
    - If inactive, pools may be temporarily reduced to 256K (except the machine and base pool).
5. The maximum pool size for an active pool has been increased to 3072K for each active job in the pool (up from 2048K).

3.12 Disk

In most commercial applications, disk activity is probably a major part of the overall response time. The following factors affect disk performance:

- DASD hardware speeds - I/O Processor, DASD controller, and disk device
- Arm utilization
- System paging activity
- Synchronous versus asynchronous disk operations
- Amount of data per access arm
- Access time
- Journaling
- Checksum
- Disk read, write caching
- RAID-5 protection
- Mirroring
- Separation of data into user auxiliary storage pools
- Fragmentation

The amount of time that a job or transaction waits for disk I/O depends on the number of other jobs in the system also doing disk I/O, and the type of operation (read, write, checksum, journal).

If requested data is not in main storage or data is to be written, a delay can occur for either a page fault or a database I/O request.

Journaling, commitment control, mirroring, and RAID-5 cause additional disk I/O operations that ensure data integrity. The performance impact of these options should be considered as a trade-off to integrity and availability. Mirroring has the least impact on disk overhead due to the increased levels of redundancy.

RAID-5 support requires additional writes to the disks, but not one-for-one as mirroring does. With the newer IOPs (internal 6502 and external 6501) and the new 9337 controller, the performance impact of RAID-5 is minimal when compared to the older disks, such as the 9337-0x0 and 9336 models.

The Performance Tool reports provide job disk I/O count values for both synchronous and asynchronous I/O operations for both database and non-database reads and writes.

3.12.1 Synchronous and Asynchronous I/O

Synchronous disk I/Os occur when the job that issued a disk I/O request waits for the operation to complete before continuing. For example, a database read by key is a synchronous operation, and job processing waits until the data is in main storage.

Asynchronous disk I/Os refer to operations where the job does not necessarily wait for the disk I/O operation to complete. Asynchronous disk operations allow the job to continue processing while the operation is in progress. For example, a sequential read (read next) often results in disk operations to get additional data into memory that are overlapped with the processing of the job.

User programs and system functions initiate disk I/O activity, but the SLIC determines whether a disk I/O function is to be synchronous or asynchronous. The system optimizes performance using asynchronous disk I/O wherever possible, but implementation in this area may change from release to release.

Use of Sequential Only processing increases asynchronous I/O. Processing records in the same sequence as they are stored on the disk also benefits from asynchronous I/O. However, using a FRCRATIO of 1 in a database file causes synchronous output and can adversely impact performance. If the newer disk IOPs (6501 and 6502) that provide write-cache are in use, the degradation of FRCRATIO(1) may be minimized as a result of caching the writes.

Each of the two types of data transfer between disk and memory (DB and NDB I/Os) may be either synchronous or asynchronous. Data transfer can be a read or a write. Thus, given all the different types of disk operations, eight counters are necessary to record disk activity. These are recorded in the performance monitor′s QAPMJOBS file when performance data is collected as:

- JBDBR - synchronous database reads
- JBDBW - synchronous database writes
- JBNDB - synchronous non-database reads
- JBNDW - synchronous non-database writes
- JBADBR - asynchronous database reads
- JBADBW - asynchronous database writes
- JBANDR - asynchronous non-database reads
- JBANDW - asynchronous non-database writes

Two additional counters record:

- JBWIO - asynchronous I/Os that required a job to wait:

  This field counts special I/Os, such as journal requests.

- JBIPF - page faults that occurred on current disk I/O operations:

  This field does not really count page faults. The count is actually of an asynchronous read or write that the system ended up waiting for. The work that caused the asynchronous I/O to be initiated completed before the I/O did. This rarely occurs, but can happen in a system environment where there is an excessive number of disk I/Os.

Field JBWIO and field JBIPF are not included in Performance Tools/400 reports, but can be included in user-written queries to get a more accurate estimation of the disk component of response time. Do this by adding JBWIO and JBIPF, and add that sum to the synchronous reads total count and subtract the value from the total of asynchronous reads.

Paging activity contributes to (but is not necessarily all of) the NDB (non-database) read count. A high NDB read count in a job may be caused by the way the job accesses data, the program structure, or the use of program working storage. Do not always assume, however, that the cause of high NDB read counts in a job is within the job. In some cases, high paging in one job is due to another job′s high main storage usage.

An asynchronous disk I/O request requires some processing by the AS/400 storage management tasks. In fact, each disk I/O operation has a cost in CPU time in addition to contributing to disk busy time.

Some system task disk I/O can be caused by writing error log or LIC log entries. If there is a lot of error logging, disk activity increases and disk queuing and high storage pool (memory) fault rates can affect performance. While this normally does not last for sustained periods, it can cause observable and sometimes significant performance degradation for short periods of time (minutes). The WRKSYSACT command, PRTCPTRPT command, PRTTNSRPT command, and STRPFRMON command can be used to determine if the error logging task (ERRLOG) on the VLIC log is active over extended time periods.

See the *AS/400 Diagnostics Aids Manual*, LY44-3900 (available to IBM-licensed customers only).

3.12.2 Physical and Logical Disk I/O

A physical disk I/O occurs when the system reads or writes a block of data from or to the disk. It involves the movement of data between the disk and the SLIC buffer. The average number of physical disk accesses per transaction is equal to the sum of the NDB (non-database) and DB (database) accesses.

A logical disk I/O occurs when a buffer is moved between the user program and system buffers. It involves the movement of information between the SLIC buffer and the ODP (Open Data Path).

Often, a logical disk I/O operation can be satisfied by data already residing in memory. On an average, many logical operations may be performed for any physical operation. The relationship between logical and physical disk I/O operations for a job can be seen in the Performance Tools Component Report (Job Workload Activity).

The ratio between logical and physical I/O may not remain constant; it can vary depending upon the environment, amount of file sharing, use of logical files, and may change from release to release based on implementation changes to system code.

Note: The logical DB I/O count is a buffer movement count, not a count of the number of records processed by the program. For example, an RPG or COBOL program performs blocking within the program when using SEQONLY(*YES number-of-records) on the database file. The program processes approximately ″x number-of-records″ for each increment of the logical DB I/O count.

If the sequential file blocking and file size remain constant from one run to the next, the logical I/O count is the same in each run. On the other hand, the number of DB (database) and NDB (non-database) I/Os can vary between runs due to differences in the execution environment.

When considering the differences in physical and logical disk I/O counts, keep the following information in mind:

1. If records are accessed sequentially, generally multiple records are read in each physical access.
2. If records are accessed sequentially by key but the records are randomly ordered in the physical file, there may be one physical access for each record for an index page.
3. If records are accessed randomly by key, there can be two to four physical accesses per record, depending on the size of the index and how many index pages need to be accessed to find the record.
4. Updating records causes more physical accesses since the changed data must be written back to disk.
5. Journal, commitment control, and checksum cause increases in the number of disk writes. Checksum also causes an increase in the number of reads.

   Note that using an Auxiliary Storage Pool (ASP) with only journals in it can significantly reduce the impact of journaling disk I/Os.
6. The system attempts to pre-read data anticipating what the program does next. Thus, some physical disk I/O operations may be asynchronous and not necessarily affect job run time.

The ″Disk Utilization″ section of the Performance Tools/400 System Report includes physical disk I/Os for each disk arm under the *Op Per Second* heading. Note that the values shown on this report represent the physical disk operations issued by the SLIC. The actual number of operations issued to a specific disk can be less than the value shown when the disk is on an IOP or disk controller that supports the ″write cache″ capability. See the description of the 6502 internal disk IOP and the 6501 external disk IOP for more information on write cache.

3.12.3 Disk I/O Processor Utilization

Disk I/O processor utilization, reported in the Performance Tools/400 Component Report ″IOP Utilizations″ section, historically has not been a bottleneck compared to average disk utilization per disk arm attached to the IOP. However, on some larger systems where RAID-5 protection is being used with large capacity disk drives (for example, 4GB) and there is more than 10 on one IOP, care should be taken when adding additional arms. BEST/1 Capacity Planner should be used to model the effects in this situation.

Based on the laboratory analysis of disk IOP utilizations from customer performance data, V3R6 Performance Tools/400 BEST/1, and the Advisor support, use the DISK IOP ″guideline″ (70%) value and ″threshold″ (80%) value.

3.12.4 Disk Arm Utilization - Percent Busy

The percent a disk arm is busy is the key disk measurement for evaluating disk performance. Disk ″utilization″ and ″percent busy″ are used interchangeably in the performance tools documentation, including this redbook. If the utilization becomes higher than an average of 40% per disk arm for all disks (except the single disk arm systems), the queuing on each disk arm begins to degrade performance. The single disk arm system is considered a special case. A single disk arm system with only one or two users can sustain peak activity at the 50% to 55% range before performance noticeably degrades.

Given a ″performance only″ basis for a decision, a two disk arm system is able to sustain higher peak system workloads than a single arm system. However, in most cases where ″cost only″ is a consideration, a single arm ***single user*** system can deliver quite acceptable performance for a rather heavy workload up to the 90% busy range.

To determine disk busy usage, you use the performance monitor to collect disk I/O operations data and use the sample reports to view the data.

The Performance Tools/400 System Report, ″Disk Utilization″ section lists percent busy under the ″Percent-Util″ column heading and the disk I/Os per second under the ″Op Per Second″ column heading.

Note that as the number of active jobs increases, queuing on the disks for data transfer can begin to degrade performance.

The concepts of Queuing Theory apply to disk actuator utilization. Therefore, the disk service time component of a transaction is affected by the number of disk I/Os, the service time of the device, and the utilization of the disk. For example, if 20 disk I/Os on a 9332 take 0.75 seconds at a utilization of 10%, it increases to 1.00 second if the utilization increases to 40%.

Table 7 on page 70 provides a table of test results for various disk hardware features at 40% busy. This table shows physical disk I/Os per second rates and typical disk service times for most AS/400 supported disks.

The newer disk controllers (external 6502, internal 6530) provide significant write cache buffering. Disk percent busy slightly higher than 40% may be tolerated in an environment where most disk operations are writes. See the index entries for more information on these controllers.

The Performance Tools Component report provides information on the effect of the disk read and write caching. There are three new columns in the Disk Activity section of the report labeled ″Cache hit Statistics″. The values are expressed as a percentage.

In the following discussion, the terms ″disk operation″ and ″disk command″ are used.

A ″disk operation″ is a request for a disk to move data to and from the CPU from and to the disk I/O controller. Not all requests from the CPU to the disk result in an operation being sent to the device. For example, on a read request, if the data is in the device cache, the data is returned to the CPU without having to physically read the data from the disk. Alternatively, if the data is not in the cache, a device read ″command″ has to be sent to the device to physically transfer the data from disk to the disk I/O processor and on back to the CPU.

The Component report columns are:

Device Read

Number of disk device cache read hits divided by the number of disk device read operations sent from the disk controller to the disk device.

As this value increases and approaches 100%, the majority of read requests are being satisfied by data in the disk cache and a device operation is not issued. As the value decreases and approaches 0%, most of the disk I/O requests sent to the disk result in a disk seek and read operation rather than being satisfied from the disk cache.

Controller Read

Number of disk controller cache read hits divided by the number of disk read commands sent from the CPU to the device.

As this value increases and approaches 100%, the majority of read requests are being satisfied by data in the controller cache and a disk operation is not issued. As the value decreases and approaches 0%, most of the disk I/O requests sent from the CPU to the disk controller result in a disk seek-and-read operation rather than being satisfied from the disk cache.

Write Efficiency

The difference between the number of write commands minus the number of write operations divided by the number of write commands.

As this value approaches 100%, the system is waiting a short time for a write command to finish. The data is buffered at the controller and written when the device is no longer busy. As the value approaches 0%, the cache is full and the controller must wait for a device write operation to complete before it can put the data into its cache and return control to the CPU with the command complete indication.

In general, it is quite normal to see high write efficiency values on the AS/400 disks. This is a positive performance indication as it indicates that even the synchronous disk writes are completing quickly and helping to improve overall throughput.

In general, as the values go towards zero, efficiency decreases and the time to get the data to and from the CPU increases. As the values approach one, the disk I/O requests are being satisfied at the speed of the disk controller or disk cache.

As the disk I/O efficiencies increase, the system CPU utilization goes up. The reason is that when the CPU is not totally committed, jobs are waiting for disk I/O command completion before they can continue processing. The less time they have to wait for disk I/O, the higher their use of the CPU is. When you go from non-cached disk I/O to using disk and controller caching, you find that throughput improves and the average CPU utilization increases because the jobs are doing the same amount of work in less time.

To the best of our knowledge at this writing, only the IBM non-6502 and non-6512 disk controllers support these counters accurately. There are no known OEM drives and controllers that accurately support the counters used to report the values in the Component report. The IBM 6502 and 6512 do not support the cache hit and device operation counters accurately.

3.12.5 Disk Percentage Used

This is a measure of the extent to which the available disk space is occupied by AS/400 objects. Disk space is allocated in extents of up to 32KB. With increasing disk occupancy, there is the possibility of fragmentation of the available space, reducing the number of free 32KB segments for allocation. When total disk space usage increases (80% to 90% or higher), disk I/O performance can degrade. At high disk occupancy, file extensions may not always be allocated in single contiguous disk extents.

It is not unusual for the Work with Disk Status (WRKDSKSTS) command or Performance Tools/400 System Report to show the *load source* disk to have a high (80% to 98%) percentage used value. This should not be considered as a cause for poor performance as most of the system code resides on this disk.

3.12.6 Disk Service Priority

A job′s run priority and time slice are not considered when performing physical disk I/O operations. All disk I/O operations are performed on a first-in-first-out (FIFO) basis. However, a job or system task′s priority does affect the speed at which the job can issue I/O operations to the system disk I/O support. That is, a batch job at a high priority can deliver more disk I/O requests to the system per unit of clock time than another job running at lower priority.

3.12.7 Disk Service Time Equation

The following example discusses the impact of disk queuing. The average time to service a physical disk I/O request can be estimated using the following equation:

```
DS = QM * DT                ; Disk service time (sec/I/O)
QM = 1 / (1 - U*((n-1)/n)   ; Affect of queuing

DT = d * 0.030              ; Base disk service time (sec/I/O)
                              (see table below for typical disk
                              service time values)

      n = number of users
```

```
d = number of disk I/O per transaction
U = UTIL * 0.01
UTIL = average disk busy, measured in percent.
```

Example

Using the preceding equations, a single disk operation performed on a disk busy 30% of the time with six users takes 0.039 seconds on an average.

```
Assume d = 1 , n = 6 and U = 0.30

DT = .030

QM = 1 / (1 - U*((6-1)/6)
QM = 1 / (1 - 0.30*(5/6))
QM = 1 / (1 - 0.30*0.83)
QM = 1 / 0.751
QM = 1.33

DS = 1.43 * 0.030 = 0.039 sec
```

The preceding formula does not take into account the write and read disk caches available on disks attached to the 6501, 6502, 6512, and 6530 disk controllers. There is no known formula to accommodate disk read cache hit percentages.

You can determine the disk cache write and read cache hit percentages by querying the Performance Monitor QAPMDISK file data for the DSDCRH field, DSDCPH field, DSDCWH field, and DSDCFW field. See the *Work Management Guide* for more information on these fields. See index entries in this redbook for more information on disk cache.

Table 7 on page 70 contains average disk service times that can be expected for most of the AS/400 supported IBM disks. The Performance Tools/400 System Report (Disk Utilization section) reports disk service time and other disk related statistics for the actual customer application environment.

3.12.8 6502/6512/6530 Disk Unit Controllers for RAID

The 6502, 6512, and 6530 I/O Processors (IOPs) for internal DASD have three SCSI buses capable of a total of 20MB per second. These IOPs also contain the ″controller function and storage″. On the 6501 external disk IOP, the associated controller is separate from the IOP. The 6530, 6502, and 6512 IOPs offer significant performance improvements over the older internal disk support when the new 1.03GB disk, 1.967GB disk, and 4.194GB disk are attached. The 6530 provides non-RAID support and the 6502 and 6512 support both RAID-5 and non-RAID for attached disks. Greater detail on performance of the 6606 (1.967GB) and 6607 (4.194GB) can be found in Chapter 5, “DASD Performance” on page 69.

The 6502 contains a non-volatile 2MB write-cache; the 6512 contains a 4MB write-cache and supports up to 16 disks, depending on the tower configuration. The write-cache is key to improved performance for heavy ″disk write″ environments by minimizing the number of actual physical disk I/Os. This means a disk attached to a write-cache disk controller enables more data to be processed for each actual physical disk I/O issued by the system software

(SLIC). Only the 1.031GB disk, 1.967GB disk, and the 4.194BG disk make use of the non-volatile write-cache.

In addition, each 1.031GB disk drive, 1.967GB disk drive, and 4.194GB disk drive has its own 512K buffer area for managing its own disk area.

In RAID-5 mode, the checksum stripes are spread over four disks when the disk configuration is from four disks to seven disks. With eight disks configured for RAID-5, the checksum stripes can be spread over all eight disks for greater efficiency.

Table 7 on page 70 contains performance summary information for disks attached to the 6502, 6512, and 6530 controllers.

Note: The internal 6530 IOP does not have write cache and does not provide RAID-5 protection.

3.12.9 6501 External I/O Processor

The 6501 IOP has a separate disk controller and supports only the 9337 DASD family of disks. Both the 9337-2xx DASD and the 9337-4xx DASD are supported. This IOP has two SCSI buses capable of 10MB per second between the IOP and system, and the IOP and controller.

With the October, 1994 announcement of the 9337-4xx DASD, a new, faster controller is available for the 6501. This new controller contains a 4MB non-volatile write-cache and supports both the 9337-2xx DASD and 9337-4xx DASD. The disk controller originally available with the 6501 supports a 1MB non-volatile write-cache.

As with the 6502 IOP, the write-cache further reduces the number of physical writes to the attached disk arms as compared to the physical disk I/O operations issued by the SLIC.

With the new 6501 controller, performance between the 9337-2xx model and 9337-4xx model is essentially the same.

Table 7 on page 70 contains performance summary information for disks attached to the 6501.

3.12.10 Disk Guidelines

This topic discusses a new metric (Physical Disk I/O Operations per Second per Gigabyte of disk storage (″Ops/Sec/GB″)). This metric can be particularly helpful when considering upgrading or adding to your current disk configuration and addresses the performance versus storage capacity concern with high data capacity under fewer disk arms.

Ops/Sec/GB is a measurement of throughput per disk actuator (″arm″). Since DASD devices have different capacities per actuator, operations per second per GB is used to normalize throughput for different capacities.

As previously discussed, a stable operating environment is when the average arm percent busy is less than 40% at peak workload periods. For each DASD model, an Ops/Sec/GB range has been established such that if DASD subsystem performance is within this range, the average arm percent busy meets the guideline of not exceeding 40%. The Ops/Sec/GB value plus the amount of

DASD space required allows you to select the acceptable DASD models to add or upgrade.

You must first use the System Report Disk Utilization section to analyze your current Ops/Sec/GB for the group of the same disk models (for example, internal 2800s, 9336, 9337-025, 9337-220, and so on) that you plan on replacing or want to use as a ″current Ops/Sec/GB base″ for replacing or adding disks. Then consider workload growth rates over the calendar time you plan for your new disk configuration. You can compare the Ops/Sec/GB rates shown in this section for the various internal disk and external disk models available.

While the method discussed in this section works well, you are reminded that the Performance Tools/400 BEST/1 provides much more detailed DASD performance analysis and capacity planning tools in selecting a new disk configuration that meets customer workload growth.

Chapter 5, “DASD Performance” on page 69 provides a sample Disk Utilization report (Figure 19 on page 85) in a scenario that shows how to select the appropriate new disk models if you presume little or no increase in disk I/O activity.

3.13 Communications

Communications line time may be defined as the time delay in transmitting the data to and from the device to the system. Contributors to the total line time include:

- I/O Processor (IOP)
- Line
- Control unit
- Device

The control unit must wait to be polled by the system prior to transmitting data. This can result in a delay, which is dependent on the amount of traffic on the line and controlling unit. For example, if you have printers and workstations on the same line, a degradation may result during the operation of the printer.

High IOP, line, and controller utilization can impact response times. Note that some communications IOP activity results from polling in addition to that caused by normal line activity.

3.13.1 Communications Errors

Most communications errors are handled by the Communications I/O Processor (CIOP). This can increase the IOP utilization. However, if the error logging occurs to the Communications Error Log or the SLIC Log, system performance may be affected because error logging runs at a higher priority than user jobs and, in some cases, uses significant amounts of CPU time generating the log records. This may result in a noticeable degradation in overall system performance and interactive applications.

Note that high rates of communications error recovery may be responsible for high page fault rates in the machine pool. You can evaluate if high communications error recovery is occurring by:

- Setting the communication line description THRESHOLD parameter to *MAX.

Based on each line protocol, *MAX sends messages to QSYSOPR when there are approximately 16 recoverable errors for every 256 frames sent.

- Reviewing the Performance Tools/400 System Report and Resource Report communications statistics.

3.13.2 Guidelines

The guideline for line utilization for excellent response time is less than 30% to 40%, though up to 50% is the more commonly used value. The IOP utilization guideline is typically quoted as 50% or less.

See Appendix A, “Guidelines for Interpreting Performance Data” on page 383 for more information.

3.14 Job Execution

Factors in the job execution environment have an impact on the overall performance of the system and response time of interactive transactions. This section discusses some of these considerations.

3.14.1 Dynamic Priority Scheduling (DPS)

For details on this topic, see Section 3.7, “Dynamic Priority Scheduling” on page 26.

3.14.2 Activity Levels

Jobs must occupy an activity level prior to becoming eligible for processing by the CPU. In analyzing activity level performance considerations, you need to understand that:

- You should use the recommended settings in *AS/400 Performance Tools Guide* or see Appendix A, “Guidelines for Interpreting Performance Data” on page 383 for more information.
- Changing the activity level value does not necessarily result in an immediate change in system performance.
- Too high a value can cause occasional performance degradation due to high paging; sometimes this can be severe.
- Too low a value can reduce interactive throughput and increase response times.
- Have the Wait-to-Ineligible counts just above zero.
- Activity level is really a control on pool paging and disk I/O.

A job may have everything it needs and still not be able to get the CPU. This happens when all activity levels are in use. Any more work overloads the system and causes additional paging and longer CPU and disk queuing. Overload can be minimized by using a storage pool activity level value small enough to keep page demand and page fault disk I/O at reasonable levels. A smaller activity level value ensures that no additional jobs are allowed to run in the pool until older work finishes. This also results in minimizing the number of jobs contending for the same resource causing seize/lock contention, and ″thrashing″, where the system spends its resources paging rather than processing information. See the *AS/400 Performance Tools Guide* or

Appendix A, “Guidelines for Interpreting Performance Data” on page 383 for information on setting activity level values.

3.14.3 Job Priority Adjustments

The AS/400 system has an implicit (separate from the Class Description object run priority parameter (RUNPTY)) ″priority adjust″ function that is associated with seize/lock conflict processing for database (and other objects). When conflict occurs, the priority adjust function determines if a requesting job has a higher run priority than that of the holding job. If it does, the priority of the holding job is temporarily made equal to the requesting job *until the seize/lock conflict is released*.

The effect of this temporary ″priority adjust″ is to provide the lower priority job access to the CPU to finish its work and release the seize/lock as quickly as possible. The promoted job′s priority is reset to its previous value at the end of the conflict. You may observe this priority adjustment with the WRKSYSACT (Work with System Activity) command.

Note that when a job has completed waiting for disk I/O, it is placed on the SLIC Task Dispatching Queue (TDQ) *ahead of* other jobs with the same RUNPRTY value.

3.14.4 Time Slice

Every time a job uses 0.5 seconds of CPU time (.2 seconds on the faster processors) between long waits, the system checks if there are jobs of equal priority on the CPU queue. If there are, the next job with equal priority is granted the CPU and the interrupted job is moved to the queue as the last of equal priority. The job, however, retains its activity level. This is an internal time slice end.

When a job reaches the external time slice value, there can be a job state transition from active to ineligible if another job is waiting for an activity level. When a job is forced out of its activity level, its pages are liable to be stolen by other jobs, and cause additional disk I/O when the job regains an activity level.

The IBM-supplied default values of two seconds for interactive jobs and five seconds for batch jobs may often be too high, especially for the high-end processors. As an initial value, set the time slice for interactive jobs at three times the average CPU seconds per transaction.

3.14.5 Object and Record Seize/Lock Conflicts

Note, as with most multi-tasking systems, there can be a lock-wait conflict for a single resource from time to time. On the AS/400 system, the condition is called a lock conflict if it occurs based on an operating system function. If it occurs during the execution of a complex MI instruction such as Retrieve Data Space Entry, it is called a seize conflict.

Examples of conflicts include waiting for a record that is locked for update, waiting for a data queue that is being updated, or waiting to use a database file while its index (access path) is being updated. In addition, frequent object creation and deletions, as are typical in a migrated System/36 application, can demonstrate seize delays on a user profile or library containing the object. The length of these conflicts impact performance and both seizes and locks can affect queuing of physical disk I/O. Lock and seize information is shown on the various Print Transaction Report output, such as the Job Summary Report and the Lock Report.

It is normal for these locks and seizes to occur for short periods of time, such as 0.001 to 0.01 seconds or less. It is when these lock or seize waits become long enough to cause perceptible user delays (perhaps due to a long series of small waits) that performance is degraded enough to require performance problem analysis.

In a system with high CPU utilizations (above 90%) the effect of high rates of seize/lock conflicts can degrade performance more than occurs at CPU utilizations less than 90%.

In many cases where long lock wait or hundreds of short seize/lock conflict wait conditions occur in a short period of time, application design change or job scheduling may be required.

One example where this can occur is on large systems where, in putting more work on the system, a batch job is duplicated so that multiple copies of the same job can run over different parts of a database file. If the jobs are updating the file, there are many more short seize conflicts that are normal as far as the database processing is concerned. Usually in this case, there is no large impact on performance and the jobs run normally.

UNIX-Processing Resource Conflicts

Note that with V3R6 and the AS/400 CPA Toolkit, UNIX applications may be ported to the AS/400 system more easily. UNIX has its own unique work management and resource conflict resolution techniques. One of these techniques is called a mutex. DSPJOB contains an option to display the active mutexes.

A *mutex* (**mut**ual **ex**clusion) is a mechanism for synchronization between UNIX ″threads″ (similar to AS/400 jobs). Mutexes provide a fast interface to serialize access to shared global storage.

If not used properly, they can cause ***deadlock*** situations. Additional UNIX information is beyond the scope of this redbook.

3.14.6 Activation Groups and the Integrated Language Environment

The Integrated Language Environment (ILE) was introduced to the AS/400 system beginning with V2R3 and the C/400 High Level Language (HLL). This environment is significantly different than the ″Original Program Model″ (OPM) used originally for Control Language, RPG, COBOL, and the other high level languages in the initial releases of the AS/400 system. The ILE is also a significant extension to the Extended Program Model (EPM) introduced to support the initial release of the C programming language (C/400).

Note: Proper use of ILE facilities can significantly improve the performance of modular programming applications where key ″programs″ are external to the mainline program and called repetitively within a performance measurement cycle, such as an interactive workstation transaction. This more easily enables ″multiple use programs″ to be developed and shared among several applications while achieving call performance close to calls to internal routines within a single large program. Note that performance of a single large monolithic program in ILE for V3R6 versus OPM for V2R3 should be approximately the same if no coding changes are made.

The ILE is ″UNIX-oriented″ and necessary to support the AS/400 system′s ″openness″ to UNIX, Spec 1170, and POSIX-based applications. ILE can also be used by traditional RPG and COBOL applications should their developers want to make use of the ILE formal constructs of procedures, modules, service programs, bound programs, and control boundaries.

With V3R6, Control Language, RPG IV/400, and COBOL/400 support the ILE capabilities.

From a performance management viewpoint, the ILE adds new terms and constructs that need to be understood to evaluate ″good performance design″ and determine if performance problems have been introduced because of misuse of the ILE constructs.

These ″ILE values″ appear on some of the system display command output and various performance tool output. A basic understanding of these values (constructs) is needed when evaluating ILE performance considerations.

This section provides a brief overview of key ILE terms and constructs that those interested in performance need to understand.

The AS/400 system provides several manuals that include ILE information. See the AS/400 *Publications Ordering*, SC41-4000-01, to determine what formal ILE documentation is available. The following redbooks also provide useful ILE information:

- *AS/400 Integrated Language Environment: A Practical Approach*, GG24-4148
- *UNIX to AS/400 ILE*, GG24-2438
- *Moving to the Integrated Language Environment for RPG IV*, GG24-4358

Section 12.14, “AS/400 ILE C/400 Tips and Techniques” on page 323 includes performance tips when using ILE facilities. Key performance considerations for ILE include:

- *Bound programs* can significantly improve performance if they are called frequently during a transaction.
- The use of *activation groups* within a job should be minimized.

ILE Procedures: A procedure is a group of related programming instructions that has a *named entry point* known to other procedures bound in a program. It is the programmer′s choice as to what set of code is contained within the procedure. Typically it is code that ″belongs together″ for a particular function, such as calculating credit discounting.

If used, a procedure, program, or module name appears in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Performance Explorer PEX *PROFILE option output (if requested), and the PEX *STATS option output (if requested).

ILE Modules: A module can consist of one or more procedures that work together to perform a function. These procedures are typically grouped together for easing program maintenance. A module may not run unless it is part of a *bound program*.

If used, a procedure, program, or module name appears in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Performance Explorer PEX *PROFILE option output (if requested), and the PEX *STATS option output (if requested).

ILE Service Programs: A service program provides a common set of *procedures* that **cannot be dynamically called** at run time. A service program offers an ″ease of maintenance″ option when activity on one of the included modules necessitates a source code change and re-creation. The modules they contain can be changed without re-binding the bound program under certain conditions. See HLL specific documentation for details.

One or more service programs are ″bound″ to a mainline program through the Create Program command for the mainline program. This is called ″bind-by-reference.″

If a procedure is called by many programs, it is a candidate for inclusion within a service program.

If used, a procedure, program, or module name appears in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Performance Explorer PEX *PROFILE option output (if requested), and the PEX *STATS option output (if requested).

ILE Bound Programs: A bound program is a set of one or more *modules* and *service programs* and is the ″runnable program″ in the ILE. The modules are ″bound-by-copy.″ Within the bound program, the addresses of the modules are ″resolved″ at the end of the creation process. At run time, this makes the ″call bound program″ function **significantly faster than the OPM ″dynamic call program″ function (even compared to OPM RPG and COBOL compiler-generated code that ″remembers″ the address of the previous dynamically called program)**.

If used, a procedure, program, or module name appears in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Performance Explorer PEX *PROFILE option output (if requested), and the PEX *STATS option output (if requested).

ILE Program Entry Procedure (PEP): A PEP is the first procedure name placed on the call stack following a dynamic call. The PEP ensures that the procedure name the programmer specified as the *entry module* on the CRTPGM command is given control following a dynamic call.

Service Programs never have a PEP. This is because a service program is merely a packaging of procedures that may be called by a main program or procedure.

The name of the PEP placed on the call stack depends on the ILE HLL used for the entry module of the program. These names may be:

- _C_PEP for ILE C/400
- _Q_QRNP_PEP for ILE RPG/400
- _CL_PEP for ILE CL
- _QLN_PEP for ILE COBOL/400

ILE Control Boundary: A control boundary is a job call stack entry used as the point to which control is transferred when an unmonitored error occurs or a High Level Language ″termination verb/operation″ is issued. A control boundary acts as a delimiter for:

- A run unit in ILE languages
- ILE exception message percolation:

  An exception that is not handled by lower level programs or modules in the stack is ″percolated″ up to the control boundary stack entry.

A procedure is a control boundary if one of the following is true:

- The caller is an OPM program.
- The caller is running in a different activation group.

There are more considerations for understanding ILE control boundaries, but they are beyond the scope of this redbook. See the redbook *Moving to Integrated Language Environment for RPG IV*, GG24-4358, and the appropriate HLL manual for a complete discussion of exception handling and control boundaries.

From a performance viewpoint, the performance impact in the OPM environment of an unhandled exception is noticeable. In the ILE, the percolation of an unhandled exception requires more system overhead. CL and RPG handling of exceptions in either the ILE or OPM environments is similar. For example, RPG file information data structure, error indicators, the file I/O INFSR subroutines, and *PSSR subroutines should still be used. For ILE C/400, there are no HLL specific handlers provided. So C/400 programmers should code specific exception handling routines.

Activation Groups: An **activation group** is a construct or ″job abstraction″ for partitioning programs and files opened by those programs within a job. An activation group is considered a ″sub-job″ where its variables and files are portioned from usage by other programs within separate activation groups in the same job. Programs within the same activation group can interact with each other within that group.

Activation groups were introduced with Version 2 Release 3 for the Integrated Language Environment (ILE) support for C/400 called ILE C/400. ILE RPG/400, ILE COBOL/400, and CL all supported activation groups with their Version 3 Release 1 availability and are now supported in Version 3 Release 6.

Programs within a job and files opened by that job are ″assigned to an activation group″ either explicitly or implicitly. Programs created in releases prior to V3R1/V3R6 default to the job *default activation group.* Programs created in V3R6 can be assigned to run within an activation group as follows:

- *DFACTGRP (applies to the entire job).
- *NEW (a new activation group is created when the program is called).
- *CALLER (when the program is called, the program is activated into the calling program′s activation group).
- Specific activation group name

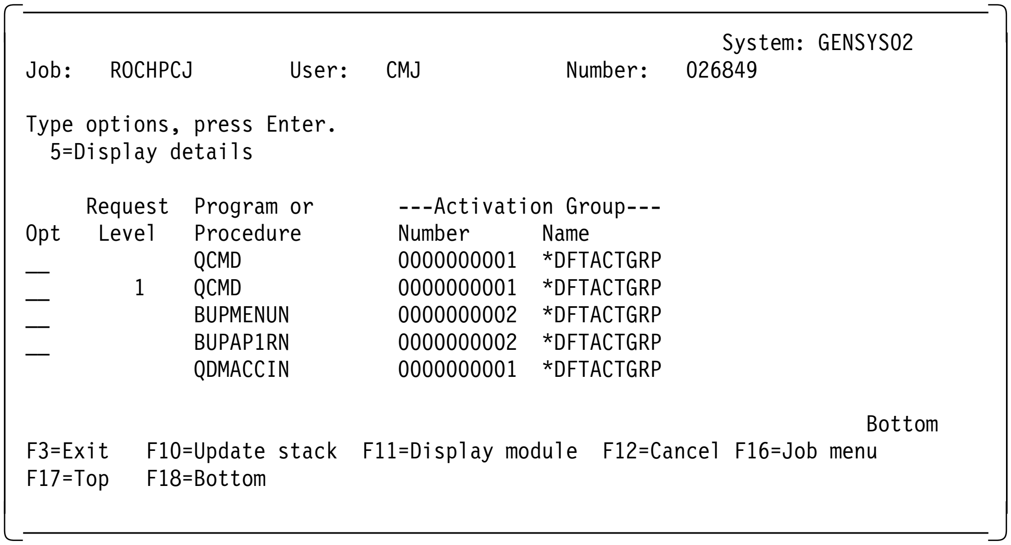
From a performance viewpoint, the creation of an activation group at run time causes a significant impact to the system (a kind of ″mini-job start′). The Create

Bound Program (CRTBNDPGM) command, Create Bound CL Program (CRTBNDCL) command, and Create Bound RPG Program (CRTBNDRPG) command contain parameters that assign the bound program to the desired activation group variable.

The Display Job (DSPJOB) command displays various ″activation group″ values even if none of the programs have been created on V3R6. Figure 8 shows an example of DSPJOB command ″display call stack and display activation group″ output for programs created prior to V3R6. Figure 9 on page 60 and Figure 10 on page 60 show similar output for ILE programs created on V3R6.

The DSPJOB ″Display Call Stack″ option can show the ILE module name, ILE program name, and control boundary program name.

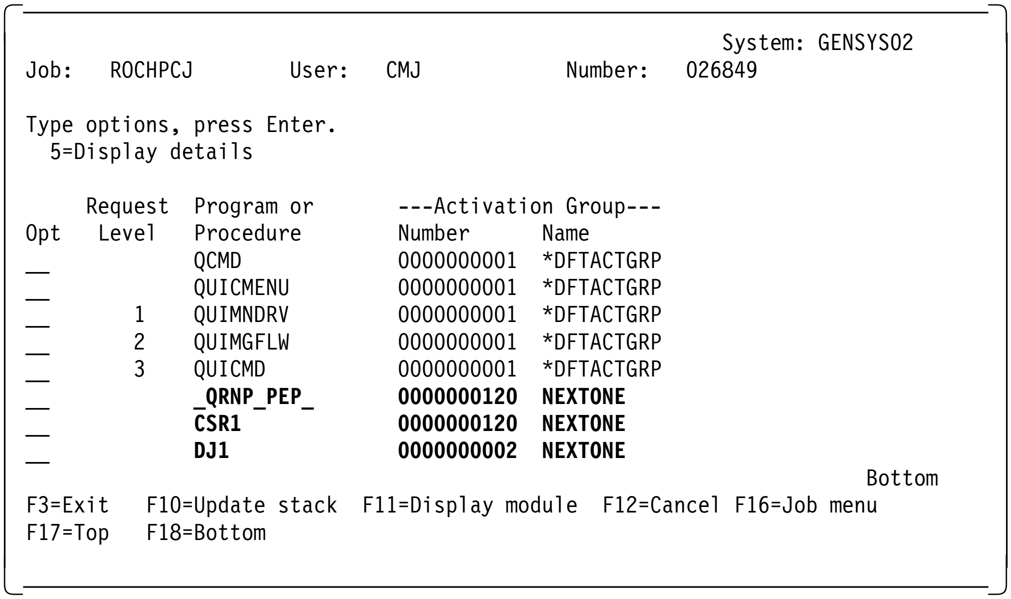
Since opened files can be scoped to an activation group, the DSPJOB command display open files option shows the opened file and its associated activation group.



```
                                                          System: GENSYS02
Job:   ROCHPCJ       User:   CMJ          Number:   026849

Type options, press Enter.
  5=Display details

     Request  Program or       ---Activation Group---
Opt   Level   Procedure        Number      Name
__            QCMD             0000000001  *DFTACTGRP
__        1   QCMD             0000000001  *DFTACTGRP
__            BUPMENUN         0000000002  *DFTACTGRP
__            BUPAP1RN         0000000002  *DFTACTGRP
              QDMACCIN         0000000001  *DFTACTGRP

                                                                       Bottom
F3=Exit   F10=Update stack  F11=Display module  F12=Cancel F16=Job menu
F17=Top   F18=Bottom
```

Figure 8. DSPJOB Call Stack Activation Group Information - No ILE Programs

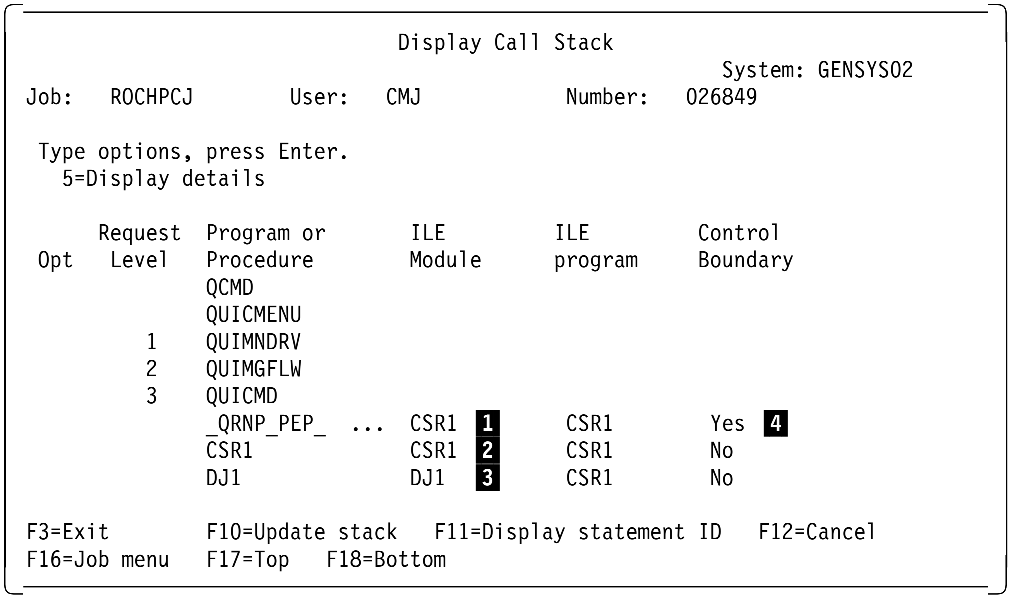
When a new job is started, the default activation group (*DFTACTGRP) is actually two activation groups. ″0000000001″ is reserved for running system functions. ″0000000002″ is the default available for application programs.



```
                                                           System: GENSYS02
Job:   ROCHPCJ       User:   CMJ           Number:   026849

Type options, press Enter.
  5=Display details

     Request  Program or       ---Activation Group---
Opt   Level   Procedure        Number      Name
__            QCMD             0000000001  *DFTACTGRP
__            QUICMENU         0000000001  *DFTACTGRP
__       1    QUIMNDRV         0000000001  *DFTACTGRP
__       2    QUIMGFLW         0000000001  *DFTACTGRP
__       3    QUICMD           0000000001  *DFTACTGRP
__            _QRNP_PEP_       0000000120  NEXTONE
__            CSR1             0000000120  NEXTONE
__            DJ1              0000000002  NEXTONE
                                                                     Bottom
F3=Exit   F10=Update stack  F11=Display module  F12=Cancel F16=Job menu
F17=Top   F18=Bottom
```

Figure 9. DSPJOB Call Stack Activation Group Information - ILE Programs

QRNP_PEP is the RPG High Level Language specific *Program Entry Point (PEP) main entry point* and does not identify the actual user-written program/module/procedure name. You must use **F11=Display Module** to determine the actual module and program called. Figure 10 shows this information for the RPG example.



```
                               Display Call Stack
                                                           System: GENSYS02
Job:   ROCHPCJ       User:   CMJ           Number:   026849

 Type options, press Enter.
   5=Display details

      Request  Program or       ILE         ILE         Control
 Opt   Level   Procedure        Module      program     Boundary
               QCMD
               QUICMENU
          1    QUIMNDRV
          2    QUIMGFLW
          3    QUICMD
               _QRNP_PEP_  ...  CSR1  1      CSR1        Yes  4
               CSR1             CSR1  2      CSR1        No
               DJ1              DJ1   3      CSR1        No

F3=Exit        F10=Update stack   F11=Display statement ID   F12=Cancel
F16=Job menu   F17=Top   F18=Bottom
```

Figure 10. DSPJOB Call Stack - ILE Modules

Notes:

1 This shows ILE program CSR1 was dynamically called.

2 For ILE RPG/400 and ILE CL, this procedure name is always the same as the module name. For ILE C/400, there can be many procedures in one module.

3 Notice that there is no PEP to identify the HLL of procedure DJ1 as it was executed by a call bound.

4 Notice the control boundary that has resulted from calling the program CSR1 that was created to run in activation group NEXTONE.

3.15 Non-Interactive or Batch Jobs

Non-interacitve jobs are sometimes overlooked in a performance and throughput review. As previously discussed in Section 4.1.2, “Determining What Job Types are Considered ″Interactive″” on page 65, non-interactive jobs can be identified by the absence of an ″I″ in the QAPMJOBS database file JBTYPE field.

The major factors that affect a non-interactive job that does not ″wait″ for work to do are:

- Job characteristics:
  - CPU usage
  - Disk I/O activity
  - Job priority
- System environment:
  - CPU usage of equal and higher priority jobs
  - Total disk I/O activity
  - Number of disk actuators
  - Whether it shares a pool with other jobs or is in a dedicated pool

If the non-interactive job is running in a dedicated environment, it is much easier to estimate the workloads and, therefore, to make a prediction on the affect of increased workloads and changes in hardware.

The remainder of this section discusses a traditional ″batch job″ and a way to evaluate performance based on the number of records processed.

An indicator of non-interactive job throughput is the number of logical database I/O (LIOs) operations per second. If the job processes the same data in multiple runs, and the programs and file blocking are not changed, the logical disk I/Os are the same for the different runs. However, the physical I/Os can vary between runs due to other jobs in the system that may cause page stealing and so on.

When a non-interactive job runs in a dedicated environment, it may be assumed for simplicity, that there is no resource contention. Therefore, the only components contributing to elapsed time once the job starts to run are:

- CPU time
- Disk I/O time

A simple formula for batch job run time is shown in the following example. An example of estimating job time from this formula is contained in Section 8.11, “Manual Batch Run-Time Estimation” on page 124.

```
Batch job run time = CPU time + Disk I/O time
                   = (LDIO x CPU/LDIO) + (PDIO x DRES)
                   = (LDIO x CPU/LDIO) +(LDIO x PDIO/LDIO) x DRES))
                   = LDIO x (CPU/LDIO + (PDIO/LDIO x DRES))


where LDIO = total logical disk I/Os
      PDIO = total sync disk I/Os
      DRES = disk response time
```

Note: PDIO is the sum of the fields from file QAPMJOBS (JBDBR + JBNDB + JBNDW + JBDBW + JBIPF + JBWIO). The last two fields are not reported in any of the performance reports, and you have to get them by using either QUERY or SQL. In many cases, the values in JBIPF and JBWIO are rather small and in others they can be quite large; looking at the values with query or other reporting function is the only way to verify this.

The Performance Tools System Report provides average values for this information. A query on Performance Monitor file QAPMJOBS can assist you in getting these values for specific batch jobs.

Note that some non-interactive jobs wait for work, such as a record to be added to a database file or queue entries on a message queue or data queue. If there are performance concerns with these types of jobs, they must be analyzed similar to an interactive application even though the system classifies them as non-interactive work.

Chapter 4. AS/400 Performance in a Server Environment

Since its introduction in 1988, the AS/400 system has evolved to meet customer and industry requirements. This has resulted in increased functional capacity and connectivity of the system. In V3R6, IBM has continued the AS/400 distributed client/server strategy, allowing customers to build on the existing strengths of AS/400 hardware and software, and giving customers a clear direction for the future.

In V3R6, new server models were announced that significantly improve system performance for both client/server and interactive workloads.

In V3R1, IBM announced support for the File Server Input/Output Processor (FSIOP), which is an integrated PC server using IBM LAN Server technology. It comes with token-ring or Ethernet LAN support, and requires LAN Server for OS/400. The Integrated PC Server (also known as File Server Input/Output Processor and FSIOP) takes advantage of the AS/400 system′s disk technology, and is fully supported in V3R6.

The choice between AS/400 Advanced System models and AS/400 Advanced Server models is based on the nature of a customer′s applications and the environment in which they operate. Given a particular customer situation, there is often more than one AS/400 solution. Generally, the solution becomes a function of price/performance, growth, and adaptation to a customer′s changing requirements. Planning is essential to match these requirements to current and future technologies.

The following sections cover how to correctly position the AS/400 Advanced Server models and the Integrated PC Server, especially in relation to performance.

4.1 Positioning the AS/400 Advanced Server

The performance characteristics of AS/400 server models work in favor of client/server workloads (may also be referred to as ″non-interactive″ or ″batch″) at the expense of interactive workloads. Hence, for the same investment, a customer can expect to see better throughput for client/server and batch work, and less throughput for interactive compared to a traditional AS/400 model.

Server model performance characteristics are more suited to a number of customer environments:

Table 5 (Page 1 of 2). AS/400 Server Models - When to Choose

Model	Customer Environment
AS/400 Advanced Server	Client/Server Online Transaction Processing (OLTP) applications
	Client/Server Executive Information System (EIS), using Client/Server end-user query products
	AS/400 Application Development (especially for batch, such as compiles submitted to batch)
	Long running commercial batch applications

Table 5 (Page 2 of 2). AS/400 Server Models - When to Choose

Model	Customer Environment
AS/400 Advanced System	Workload is largely interactive using non-programmable workstations or PC′s with 5250 emulation (such as RUMBA).
	Has a large number of twinax connected devices.
	Interactive workload exceeds the maximum throughput of a server model.
Note: This table considers database server applications, not file server applications. While a server model may run applications using shared folders or IFS faster than the equivalent AS/400 Advanced System model, the best choice may be to add an Integrated PC Server to the existing AS/400 system. See Section 4.2, "Positioning the Integrated PC Server" on page 66.	

A difficult decision arises when the customer has a mixture of client/server applications and traditional transaction based applications. The relative performance rating (internal processor) of an AS/400 server model varies depending on the workload. Section A.1, "System Capacities for AS/400 PowerPC Advanced Systems" on page 384 shows the relative performance rating (RPR) when a particular server model operates with a homogeneous interactive workload and when it operates with a homogeneous non-interactive workload (that is, batch or server). Most systems operate with a combination of batch and interactive workloads. The most accurate method of estimating the affect of changing from a traditional model to a server model is to use the BEST/1 for OS/400 capacity planner. Data is captured using the Performance Monitor and BEST/1 is used to model the desired server configuration.

4.1.1 Impact of Interactive Work on Server Model Performance

As stated earlier, the server model performance is optimized for client/server and batch environments at the expense of interactive environments. This means that as interactive work is added to the server models, the ***overall system performance*** decreases. In environments where only client/server or batch work is present, the effective performance of a server model is represented by the non-interactive RPR.

However, for mixed environments, the effective performance of the server models is represented by the range of the non-interactive and interactive RPRs, depending on the amount of interactive work present and on the particular AS/400 server model. If there is only one interactive job running on the server model, its performance is at the server speed regardless of the amount of processing the job does.

Depending on the model, if the interactive CPU utilization is in the range of 2% to 10%, the overall performance is represented by the non-interactive RPR. This means that the performance of both interactive and non-interactive work is represented by the non-interactive RPR.

As the interactive CPU utilization moves above the 2% to 10% range, the overall system performance decreases for ***both interactive and non-interactive*** work until the performance of both is represented by the interactive RPR. Therefore, maximum price/performance of the server models is achieved when interactive work is kept to a minimum.

The performance varies depending on which AS/400 PowerPC server model is selected. However, the same principles apply. A SLIC interrupt handling task does additional processing for interactive work and, therefore, uses more CPU time as the amount of interactive work increases. These tasks are named CFINTn, where n is the processor number (1, 2, 3, or 4). On the server models, there is more work for this task to perform and, therefore, as the interactive workload increases, the CFINTn task (or tasks) use more of the CPU on the server models than on the non-server models.

On the performance tools system report, you see additional CPU used by the priority zero system tasks that is not there on the same report from a non-PowerPC system. Don′t panic! If it is a server model and there was interactive work running on it, this additional overhead is due to the CFINTn tasks. If it is not a server model, do more analysis.

On the performance reports that show SLIC task names (Component, Transaction, PEX STATS, and PEX TRACE), you see these additional CFINTn tasks listed.

- On the traditional models (models 400, 500, 510, and 530), the time spent in CFINTn may be only a few percent.
- On the server models, the time spent in CFINTn increases significantly as you add more non-optimized (interactive) work to the system.

  For example, when the CPU utilization for interactive work (that is, non-optimized work) on a 50S is 2%, the CPU utilization for CFINTn is only 4%. As the CPU utilization for interactive work increases to 21%, the time in CFINTn increases and CPU utilization for CFINTn increases to 40%. Finally, as the CPU utilization for interactive work increases to 24%, the time spent in CFINTn increases further and the CPU utilization for CFINTn increases to 64%. This sizeable increase in CFINTn utilization is a direct result of adding too much interactive work to the system.

  Note that all of the server models follow the same trend as far as increased use of CFINTn is concerned, but the different models have different values. For detailed predictions, use BEST/1.

To get the best price/performance from a server model, minimize the amount of non-optimized work on the system. A rule of thumb is to keep the non-optimized work on the machine to less than 2% of the total CPU utilization.

4.1.2 Determining What Job Types are Considered ″Interactive″

It is extremely important to understand whether an existing or proposed application belongs to the interactive or client/server classification in order to determine how well it performs on an AS/400 server series model. In addition, customers who buy a Model 53S, feature code 2155 or 2156, must understand the difference between speed and capacity on multi-processor systems.

The **Client Server classification** includes the following functions:

Table 6 (Page 1 of 2). Client/Server Classification

Client/Server Classification on Server Models				
Networking	**File Serving**	Batch	**Server Apps**	**Database**
APPC	Shared Folders	Compiles	Progress	DDM, Remote SQL

Table 6 (Page 2 of 2). Client/Server Classification

Client/Server Classification on Server Models				
TCP/IP		Background Jobs	Synon/CSG	ODBC, IDAPI
		Overnight Jobs	SQL Windows	DAL
			ENVY/400	File Transfer
			OMNIS/7	ShowCase VISTA

The **Interactive classification** includes the following functions:

- Non-programmable terminal (5250-type)
- PC Support/400 workstation function
- RUMBA/400
- ″Traditional″ applications (keyboard/display intensive)

An ″interactive″ job is one that uses the keyboard and a character-type display. If a job needs the user to type on the keyboard and character results are shown on the display, that job is considered ″interactive″. Interactive in this sense means that the job and the user depend on each other to get the work done.

An interactive job may start another job that does not use the keyboard or display through the SBMJOB command (or related commands). In this case, the submitted job is **not** considered interactive. However, if the activity in an interactive job is simply transferred to another program through the CALL command, the work of the new program is still considered to be interactive, even if it does not interact with the display and keyboard.

If your applications are already running on an AS/400 system, you can determine which jobs are considered interactive by the following procedure:

On a WRKACTJOB command, look in the Type column. Interactive jobs are type INT. Everything else is non-interactive.

If your system is currently running a mix of interactive and batch jobs on a traditional AS/400 model, use BEST/1 to predict what happens on a server model. BEST/1 supports the new server models and can provide estimates of both interactive and ″batch″ performance on the server models.

4.2 Positioning the Integrated PC Server

One of the strengths of the AS/400 system has been its affinity with PCs. Through the use of shared folders, the AS/400 system has been able to offer PC users the ability to use the AS/400 system as a file server. With the increase in the number of customers who have LAN attached PCs, a better performing file server is required.

For PC users who have moderate file serving needs, IFS is capable of performing adequately. However, if the AS/400 system is primarily being used for high performance file serving, adding an Integrated PC Server is essential.

While both IFS and the Integrated PC Server give users the ability to share AS/400 disk resources, the Integrated PC Server offers a significant improvement in performance, and a large reduction in the amount of AS/400 CPU resource used. In fact, the Integrated PC Server uses very little AS/400 CPU. It gives consistent performance across all AS/400 models, since the Integrated PC

Server off-loads work from the AS/400 system. See Section 13.5, “LAN Server/400 and Integrated PC Server File Serving Performance” on page 335 for more detail on Integrated PC Server performance.

When compared to the cost of a LAN adapter (such as a #2617 Ethernet adapter or a #2619 token-ring adapter) the cost of an Integrated PC Server with 32MB of memory and LAN Server/400 is not significantly more, and includes an integrated token-ring or Ethernet adapter. If required, the Integrated PC Server can be used exclusively as a LAN adapter without using its file serving capabilities.

The use of AS/400 disk resource is managed by standard AS/400 system management utilities. If high availability options such as disk mirroring or RAID-5 are configured, the Integrated PC Server automatically uses them as well. In most situations, the use of an Integrated PC Server is cost competitive with a stand-alone PC file server.

REMEMBER

All AS/400 Advanced Series models support the Integrated PC Server. The Integrated PC Server can be used (subject to the prerequisite number of slots) with all AS/400 Advanced Series models, and AS/400 D,E, and F models. An AS/400 server model is **not** a prerequisite.

Chapter 5. DASD Performance

DASD can affect performance in a number of ways. Newer technology has increased the performance of DASD while at the same time increasing density. Higher availability options are available using techniques such as mirroring and RAID-5. These may affect overall performance. This chapter compares the performance of various options to assist you in selecting the appropriate DASD configuration.

In V3R6, the page size changed from 512 bytes to 4KB. In general, this change does not noticeably alter the DASD response time characteristics of an application as long as sufficient memory is added when upgrading to V3R6 (refer to Section 3.9.6, "Memory Guidelines" on page 41 for more details). This is particularly true for batch applications that typically have I/O sizes that exceed 4KB. For interactive applications that mainly access data randomly, the 4KB page size may decrease the number of I/O operations, dependent on additional data being accessed in the same 4KB page.

5.1 Device Performance Characteristics

This section compares the performance of the internal DASD subsystems (based on the 6502 and 6512 RAID controllers or 6530 storage controller) with the external 9337 disk array subsystem using a system configured with an equivalent amount of DASD capacity. The performance is based on measurements and modeling done in a controlled laboratory environment. Because the performance of the AS/400 system is dependent on many factors, these characteristics are general in nature. To assess the various configuration options, a capacity planning tool (such as BEST/1) should be used.

The performance characteristics of internal DASD is listed in Table 7 on page 70 and the performance characteristics of external DASD is listed in Table 8 on page 71. The tables do not list all feature codes, but they do provide performance information for current disk configurations. For a description of the DASD models supported by the 6502, 6512, and 6530 IOPs, refer to the *AS/400 Advanced Series System Handbook*, GA19-5486-14.

The following measures of performance are listed in the tables:

Service Time is the time required to perform the ″Interactive operation″ described in the next paragraph. The time starts with the request from the CPU to the Disk IOP and the time stops when the data is in main storage (read) or when the data is on the disk or in the write cache (write). Queueing time is not included.

Interactive Ops/Sec is an estimate of the number of IOs that can be done at 40% utilization using the service time calculated for the previous column. If the disk model contains two arms, this number only reflects the capacity of one arm. At 40% to 50% utilization, the disk arms are at the ″knee of the curve″. As utilization exceeds the ″knee of the curve″, response time increases significantly and becomes erratic. We assume:

- 40% arm utilization
- 7KB transfer size
- 70% read and 30% write

- 80% 1/3 seek and 20% 0 seek

Interactive Rel is the relative interactive performance of the disk drives. This column is the same as the INTERACTIVE Ops/Sec column except that the numbers are normalized to 1.0.

Batch Hours is an estimate of how long a batch type application takes to run. The duration of batch type jobs often depends on the performance of the disk. For ease of understanding, the numbers are normalized to eight hours assuming the slowest disk drive is used. We assume:

- 75% of the batch job time is disk IO.
- Average of 4KB, 8KB, and 16KB transfer sizes
- 70% read and 30% write
- 20% 1/3 seek and 80% 0 seek

Ops/Sec/GB is an estimate of how many system physical disk IOs per second per usable GB of space that the specific model of DASD can perform when the arm is 40% utilized. The write cache effectiveness reduces the volume of writes that the physical disk drive must support. For the 9337-2xx models, the write cache effectiveness is assumed to be 45%, and for the 9337-4xx models, it is assumed to be 65%. For the 6502 IOP attached 1.031GB drive, 1.967GB drive, and 4.194GB drive, the write cache effectiveness is assumed to be 55%. For the 6512 IOP attached 1.031GB drive, 1.967GB drive, and 4.194GB drive, the write cache effectiveness is assumed to be 65% (write cache effectiveness figures may vary on customer systems, due to physical placement of data on the disk drives; whenever possible use actual cache effectiveness figures for capacity planning purposes). We use the service time required to physically write the record to DASD. The service time contained in column four includes the faster write completions that result when the write is safely in the write cache.

Table 7. DASD Performance - Internal DASD

Disk Model	MB	Number of Arms	Service Time	Interactive		Batch Hours	Ops/Sec/GB	
				Ops/Sec	Rel		Base	HA
6603	1967	1	14.5	27.6	1.4	4.6	14	
6602/6601	1031	1	13.8	29.0	1.5	4.5	28	
6502-6602	1031	1	10.3	38.8	1.9	3.9	32	25
6502-6603	1967	1	10.7	37.4	1.9	3.9	16	12
6502-6605	1031	1	8.7	46.0	2.3	3.6	38	29
6502-6606	1967	1	8.9	44.9	2.2	3.6	20	15
6502-6607	4194	1	9.2	43.5	2.2	3.6	9	7
6512-6605	1031	1	8.2	48.8	2.4	3.4	43	36
6512-6606	1967	1	8.4	47.6	2.4	3.4	21	18
6512-6607	4194	1	8.7	46.0	2.3	3.4	10	8
6530-6602	1031	1	13.7	29.2	1.5	4.4	28	
6530-6603	1967	1	14.2	28.2	1.4	4.4	14	
6530-6605	1031	1	11.4	35.1	1.7	3.9	34	
6530-6606	1967	1	11.7	34.2	1.7	4.0	17	
6530-6607	4194	1	12.1	33.1	1.6	4.0	8	

Table 8. DASD Performance - External DASD

Disk Model	MB	Number of Arms	Service Time	Interactive		Batch Hours	Ops/Sec/GB	
				Ops/Sec	Rel		Base	HA
9336-010	942	2	19.9	20.1	1.0	5.7	43	
9336-020	1714	2	19.9	20.1	1.0	5.7	23	
9336-025	3428	4	17.2	23.3	1.2	5.2	27	
9337-010	1084	2	19.5	20.5	1.0	5.6	38	
9337-015	1084	2	15.5	25.8	1.3	5.2	48	
9337-020	1940	2	19.5	20.5	1.0	5.6	21	
9337-025	1940	2	17.1	23.4	1.2	5.2	24	
9337-040	7868	4	17.5	22.9	1.1	5.2	12	
9337-110	1626	4	19.5	20.5	1.0	7.8		23
9337-115	1626	4	15.5	25.8	1.3	7.2		29
9337-120	2910	4	19.5	20.5	1.0	7.8		13
9337-125	2910	4	17.1	23.4	1.2	7.2		15
9337-140	5901	4	17.5	22.9	1.1	7.2		7
9337-210	1084	2	12.4	32.3	1.6	4.3	50	36
9337-215	1084	2	9.8	40.8	2.0	3.9	63	46
9337-220	1940	2	12.5	32.0	1.6	4.3	28	20
9337-225	1940	2	11.0	36.4	1.8	4.0	32	23
9337-240	7868	4	11.3	35.4	1.8	4.0	15	11
9337-420	3880	4	8.6	46.5	2.3	3.5	43	36
9337-440	7868	4	8.8	45.5	2.3	3.5	21	18
9337-480	16776	4	9.1	44.0	2.2	3.6	10	8

The newer DASD that is used in the 6502, 6512, and 6530 DASD subsystems has improved read ahead buffers that can provide performance advantages. Each of these DASD has a 512K buffer. The buffer is allocated into multiple segments that are larger than 32K each. Read ahead data from recent IOs is kept in these buffer segments. Depending on the data access patterns, it is possible that the data needed is already contained in a buffer segment. If so, no physical access to the DASD is required. Depending on your data access patterns, this can significantly improve performance. Our analysis of several specific customer installations indicates that 10% to 30% of their DASD IO for ″interactive″ transactions is already contained in the read ahead buffer. For ″batch″ type jobs, 25% to 45% of their DASD IO is already contained in the read ahead buffer. The RAMP-C workload that was used to generate the preceding tables has less than 10% of its DASD IOs already in the read ahead buffer.

For the 9337-2xx model, 9337-4xx model, and 6502/6512 attached DASD in HA mode, most batch jobs run nearly as fast as if they are run in ″base″ mode or mirrored mode. Only in extreme cases does the HA mode cause degradation. An example of the extreme case is when there are sequences of hundreds of writes to a single DASD in a short period of time.

You must ensure that you have enough arms to support the volume of DASD IOs. In some situations, using larger capacity DASD may result in an insufficient number of arms to handle the required DASD IO volume. The capacity planning tools should be used to verify any proposed changes in your configuration.

The recommended threshold for maximum DASD utilization for one-arm configurations is higher than the threshold for multiple arm configurations. The reason for the lower recommendation for multiple arms is that it is assumed that when two or more arms have an average utilization of 40%, some of the arms may be at the 50% to 55% range while others are lower. QSIZE400 and BEST/1 allow a one-arm configuration to reach 55% before they recommend that an additional DASD be added.

Consider the following example: Assume you are configuring a system and need approximately 4GB of DASD space. You have the choice of 4 x 1031MB or 2 x 1967MB. The 4 x 1031MB configuration supports approximately 70% more DASD IOs than the 2 x 1967MB configuration. Because there is a maximum number of DASD devices that can be attached to each model, using the larger drives allows more MB of DASD to be configured on your system, at the expense of less IOs per second.

The Performance Monitor (STRPFRMON) command captures additional performance data (buffer hits, and so on) for the 6502, 6512, and 6530 IOPs. This data is available in the QAPMDISK performance data file and is documented in Appendix A of *OS/400 Work Management Version 3*, SC41-4306.

5.2 DASD Performance - Interactive

This section compares the relative performance of different DASD models and DASD IOPs. Also, included are comparisons for configurations that support higher availability such as RAID-5 and mirroring.

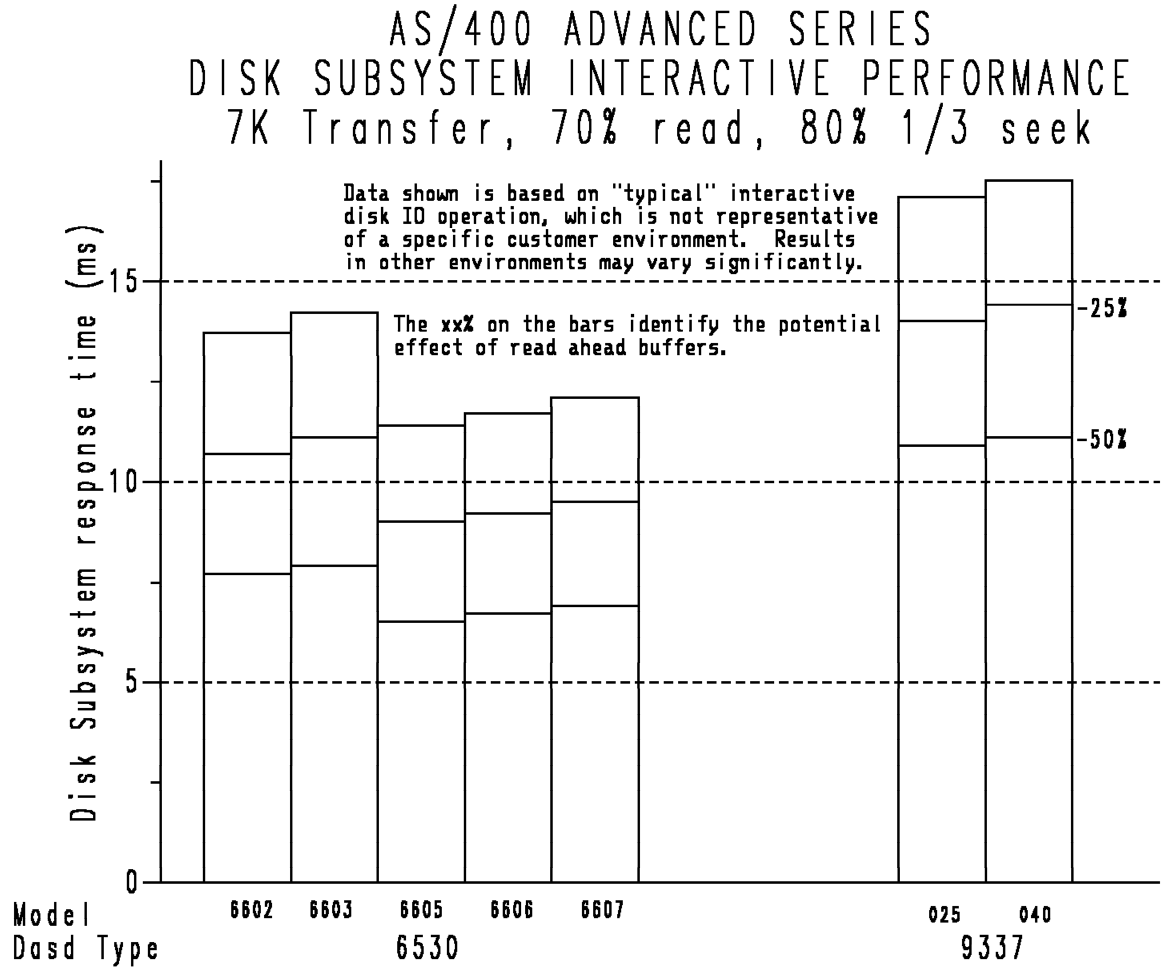
Note: These comparisons are done under synthetic laboratory conditions and are not necessarily representative of a specific customer workload. Use capacity planning tools, such as BEST/1, to assist in monitoring the impact of different DASD configurations.

The implementation of the 4KB page size on V3R6 improves system DASD IO efficiency. As a result of the larger page size, V3R6 interactive Ops/Sec/GB figures are lower compared to V3R1.

Therefore, direct comparisons between V3R6 and V3R1 DASD system performance charts are not recommended.

5.2.1 DASD Subsystem Performance - Base or Mirrored

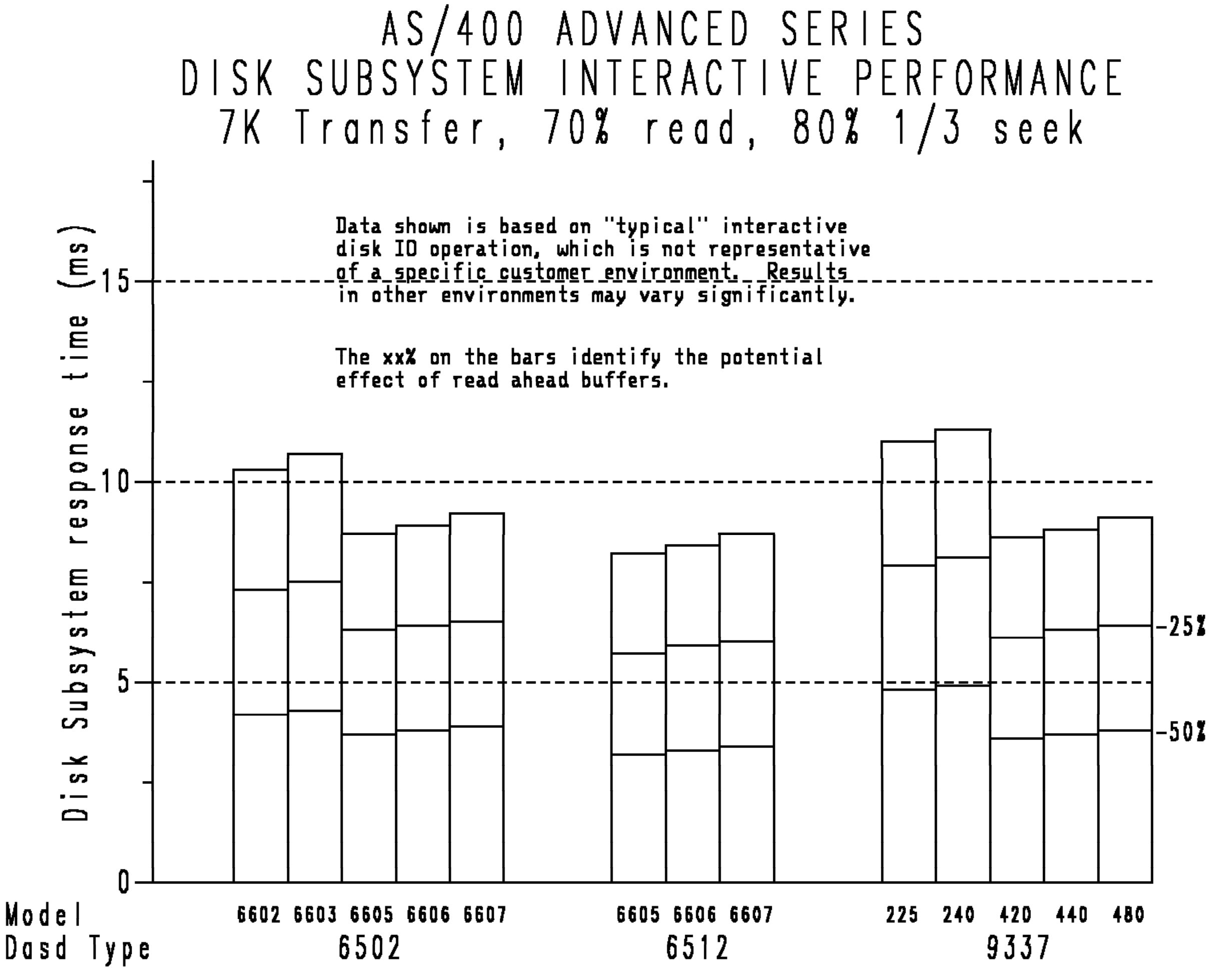
The following bar charts compare the service times for the AS/400 DASD subsystem offerings. The IO operations being performed are 7KB transfer size; 70% are reads and 30% are writes, and 80% require a seek over 1/3 of the disk surface while 20% require no seek. Queueing time is not included.



09-18-95 (D51A:CBN:R6DSD2)

Figure 11. DASD Subsystem Performance / Non-RAID Capable - Base Mode

DASD subsystem performance shown in Figure 11 and Figure 12 on page 74 give a general overview of the performance of various internal and external DASD subsystems. Certain DASD IOPs such as the 6501 (attachment of 9337 DASD subsystems) and 6502/6512 IOPs, give superior performance. This is attributable to the non-volatile write cache technology used in these IOPs. The 6512 IOP is a later (and faster) version of the 6502 IOP.



09-18-95 (D51A:CBN:R6DSD1)

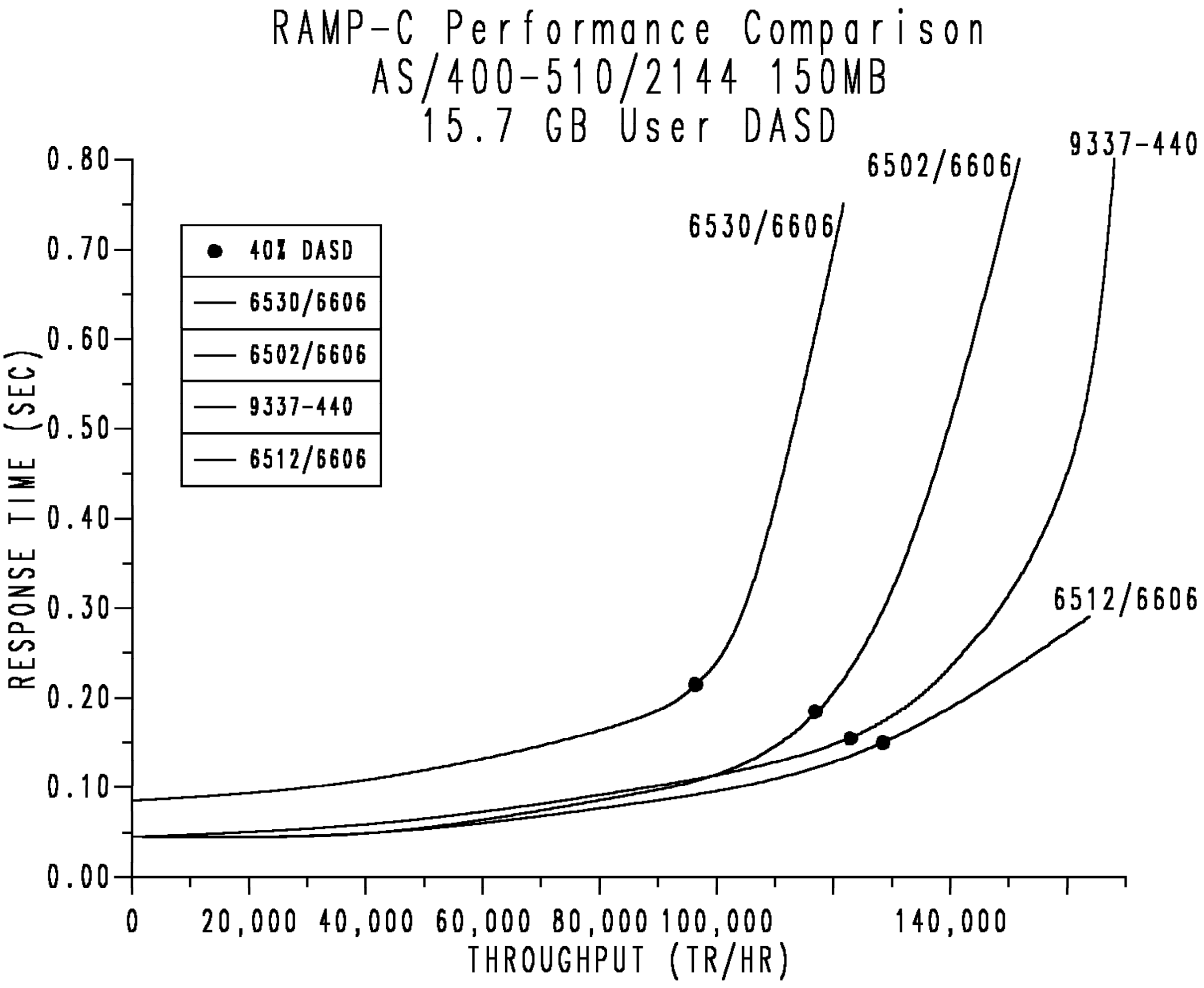
Figure 12. DASD Subsystem Performance / RAID Capable - Base Mode

The potential effect of read-ahead buffers is shown for the cases of having 25% and 50% of the total disk operations already in the read ahead buffer. Depending on the data access patterns, the buffers may provide significant performance improvements.

For batch performance results, refer to Table 7 on page 70, Table 8 on page 71, and Section 5.3, “DASD Performance - Commercial Batch (Base versus RAID)” on page 81.

5.2.2 AS/400 System Interactive Performance - Base

The following graph compares the relative interactive performance of an AS/400 model 510-2144 configured with 15.7GB of internal or external DASD. The internal load source drive was ignored for this comparison chart. The curves characterize what may occur on either a ″base″ configuration or a mirrored configuration. The graph compares the 9337-440 DASD subsystem (attached with a 6501 IOP) with the 6602 2GB DASD (attached with 6502, 6512, and 6530 IOPs).



09-19-95 (D51A:CBN:R6RCB)

Figure 13. System Interactive Performance - Base Mode

Figure 13 shows the relative throughput for various DASD subsystems. The combination of a 6512 DASD IOP and model 6606 DASD give superior performance when running a RAMP-C workload. This can be attributed to the larger non-volatile write cache (4MB) and a faster processor in the 6512 DASD IOP. A write cache greatly improves DASD service times for write operations.

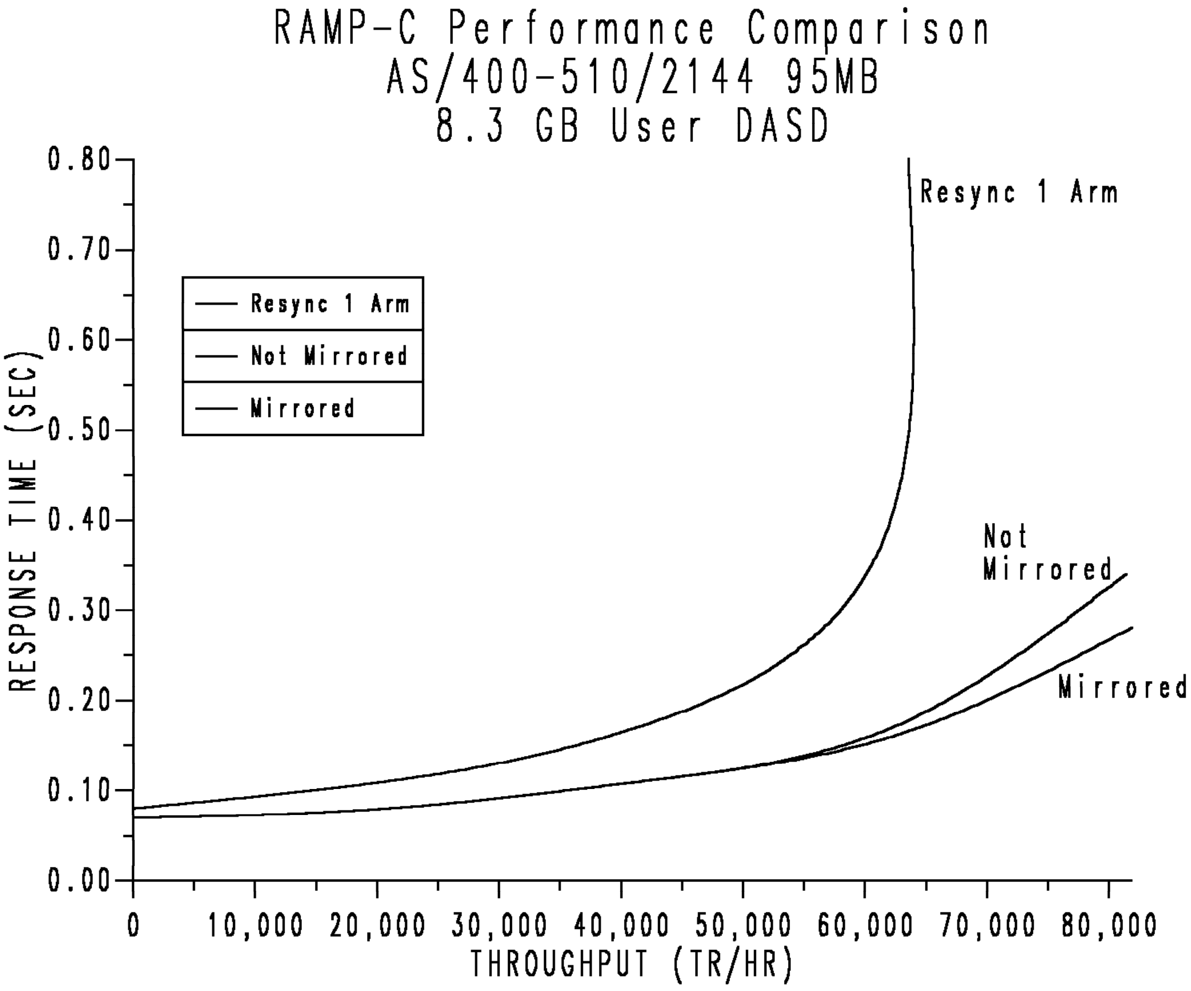
In addition, the following points should be noted:

- This graph is based on RAMP-C workload. Other environments may vary significantly.
- The RAMP-C benchmark′s data access patterns are intentionally random; therefore, the read-ahead buffers provided only minimal benefit for RAMP-C. Depending on your data access patterns, the DASD read ahead buffers may provide significant performance improvements.
- Similar results may occur on other AS/400 models. Response time and throughput curves encounter a ″knee″ when a resource is used too heavily. CPU, main memory, IOP Processor, and DASD are examples of resources that can cause ″knees″. If faster AS/400 CPUs are used, and other resources are unchanged, there is a possibility that memory or DASD can constrain the

throughput increases. The BEST/1 Capacity Planner should be used to determine appropriate configurations.

5.2.3 AS/400 System Interactive Performance - Mirrored versus Base

This section compares the relative interactive performance of an AS/400 model 510/2144 configured with 8.4GB of user DASD. The graph compares a mirrored environment with 16 arms to a base (not mirrored) environment with eight arms. It also shows the system performance effects during the resynchronization of a single arm if a disk arm fails and needs to be replaced.



09-20-95 (D51A:CBN:R6RCM)

Figure 14. System Interactive Performance (Mirrored versus Base - Internal DASD)

The performance of a mirrored configuration as shown in Figure 14 provides equal or better interactive performance than the base configuration (not mirrored). The superior performance in the mirrored configuration is due to having more arms to handle more read operations at higher throughput, since on a mirrored subsystem, a read can be performed from either side of the mirror.

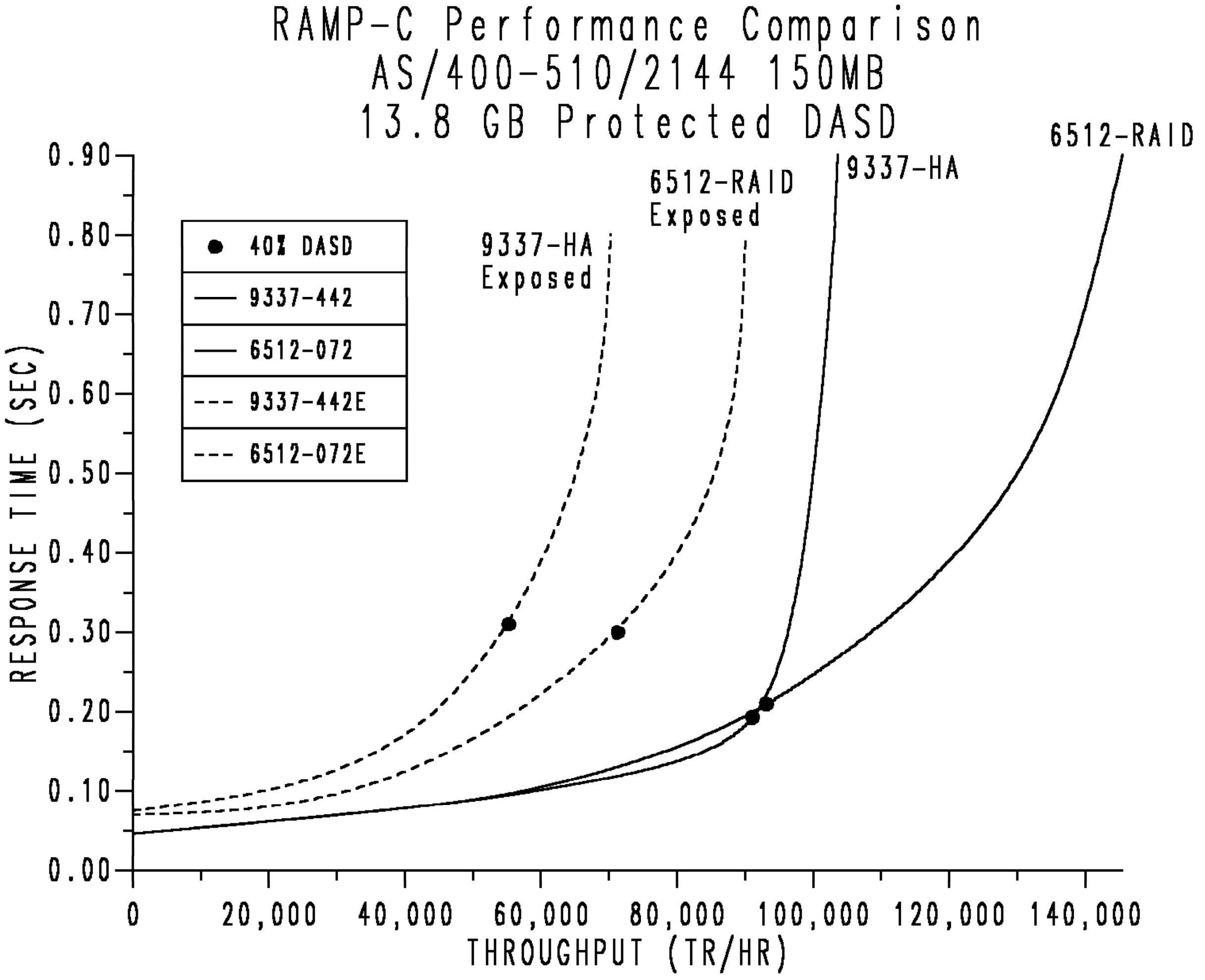
If a DASD failure should occur, the system performance is less during the time it takes to resynchronize the replacement DASD arm, especially at higher throughput. The customer can choose to schedule the resynchronization during a period of lower system activity or accept reduced system throughput during the resynchronization time (approximately 20 to 40 minutes).

Note: This graph is based on RAMP-C workload. Other environments may vary significantly.

5.2.4 AS/400 System Interactive Performance - RAID

Figure 15 compares the RAID-5 DASD subsystem (using a 6512 DASD IOP) with the 9337-440 HA subsystem (using a 6501 DASD IOP). Both subsystems contain eight 1967MB arms. Also shown, is the effect on throughput, should a DASD fail in a RAID-5 subsystem.

The 9337 measurements were done with four parity arms per array and the 6512 measurements were done with eight parity arms per array. In general, eight parity arms per array provides better performance at higher throughputs. At low to medium throughput, there is little performance difference between four and eight parity arms per an eight-arm array. On the 6512, the parity array should be configured with eight parity arms if possible.



09-21-95 (D51A:CBN:R6RCE)

Figure 15. Performance Impact of Failed DASD

Impact of a Failed DASD in RAID Subsystem

READ To read from a failed DASD, RAID-5 must read **all** remaining arms in the set (this means anywhere from three to nine overlapped reads, where one was sufficient before). This has a significant effect on the failed DASD subsystem throughput and response time. This degraded mode lasts until the DASD is repaired and rebuilding of the failed DASD′s parity stripes is complete.

Reads to other DASD on the same subsystem are unaffected.

As a guide, if the parity array has four arms, this results in 1.5 times increase in DASD IO read volume to this array. If the array has eight arms, the result is 1.75 times increase in DASD IO read volumes to this array.

WRITE There are three separate scenarios that apply to RAID-5 writes with one failed DASD:

1. If the failed DASD is not involved (either for data or for the checksum stripe), the writes are handled as normal RAID-5 writes (two reads plus two writes).
2. With a write to a failed DASD, all remaining DASD in the set must be read and one write is done to the checksum stripe (N-1 reads plus one write, N=number of DASD arms).
3. If the DASD that contains the checksum stripe is the failed DASD, all that is required is a write to the DASD that contains your data (one write).

As a guide, if the parity array has four arms, each write averages a 3.25 increase in DASD IO write volume to this array. If the array has eight arms, the result is a 4.13 times increase in DASD IO write volumes to this array.

General Discussion
The DASD IO to any subsystems that do not have a failed DASD are unaffected.

If the Customer cannot tolerate the temporary performance degradation that occurs with a RAID-5 DASD failure, they should consider mirroring.

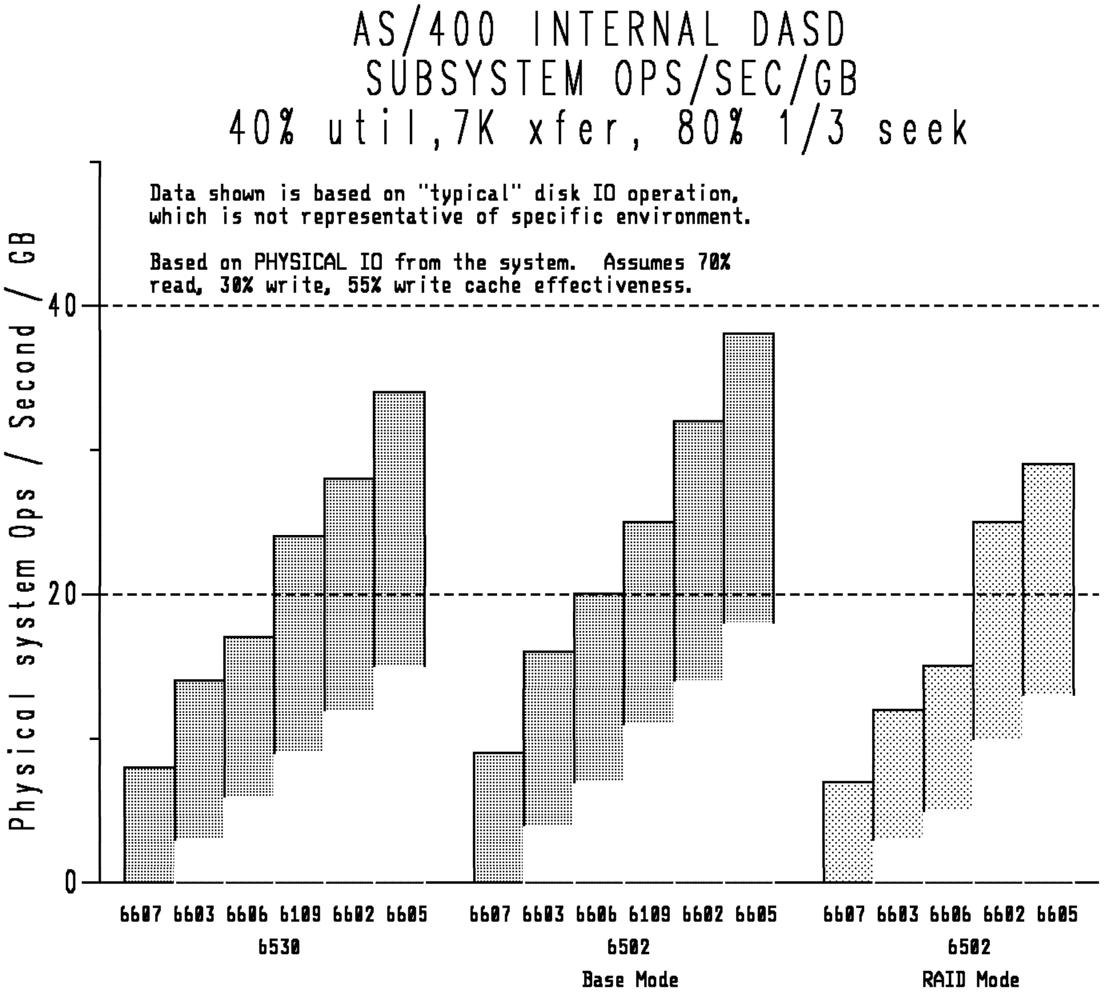
To obtain acceptable performance with a failed RAID-5 DASD, some customers may have to delay nonessential work until after the DASD is repaired. For example, a customer may continue to process their online order entries but delay their office tasks.

The estimated time to rebuild a DASD is approximately 30 minutes for an eight-arm array on a dedicated system with no other jobs running. If other concurrent jobs being run on the system are requesting, for example, 130 IOs per second to this DASD subsystem, the rebuild time increases approximately one hour.

5.2.5 Ops/Sec/GB Guidelines for DASD Subsystems

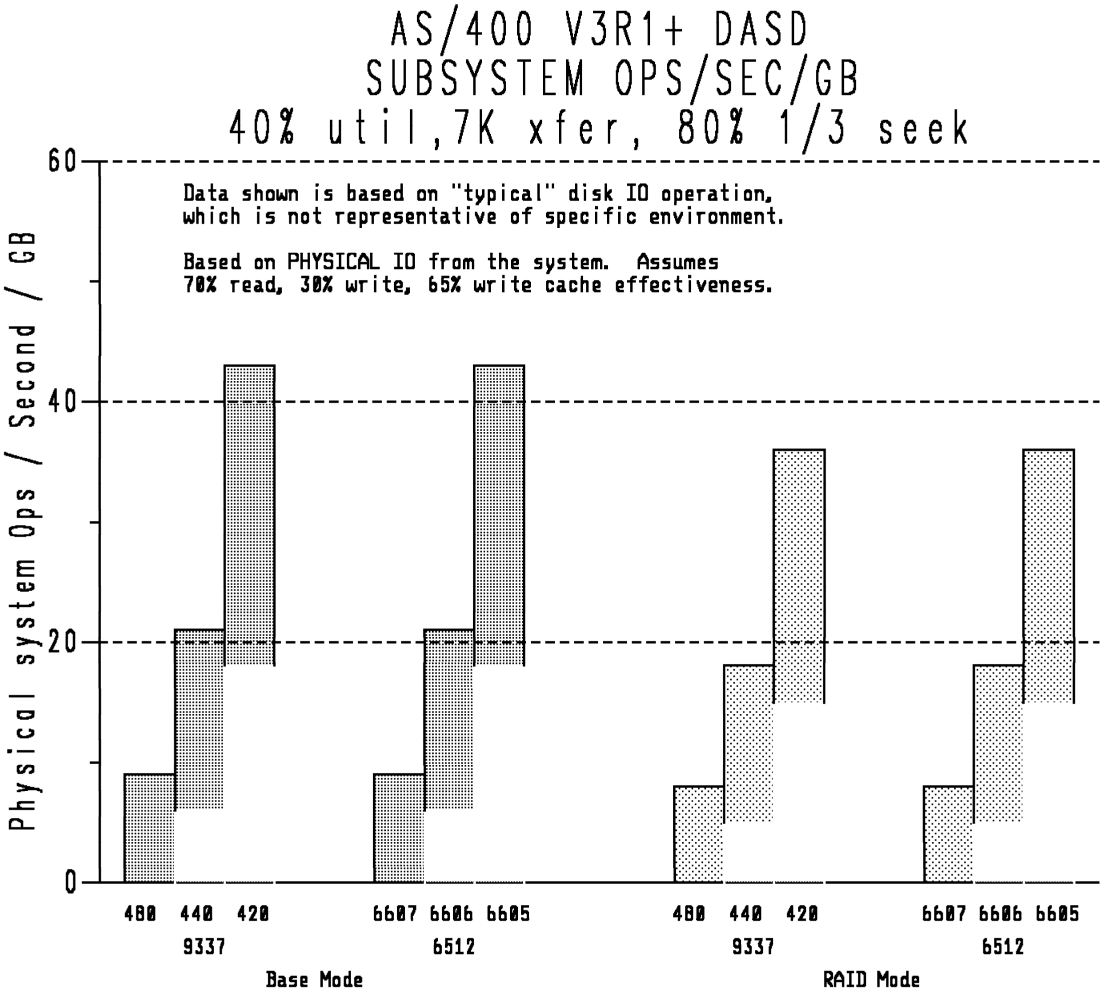
In this section, the concept of Ops/Sec/GB charts is used to provide a guideline in the choice of appropriate DASD models when adding or upgrading DASD. In conjunction with these charts, you should utilize results obtained from the Performance Tools reports to determine which DASD models meets your DASD performance requirements. For more detailed DASD performance analysis, it is recommended that a capacity planner such as as BEST/1 be used.

Operations per second is a measurement of throughput per actuator. Since DASD devices have different capacities per actuator, operations per second per GB is used to normalize throughput for different capacities.



09-12-95 (D51A:CBN:R60PS2)

Figure 16. Ops/Sec/GB - Internal DASD



09-12-95 (D51A:CBN:R6OPS1)

Figure 17. Ops/Sec/GB - Internal versus External DASD

To determine the operations per second of your current operating environment, use the following procedure. The value obtained by this procedure helps determine what DASD model meets your current or projected DASD performance requirements.

1. Collect performance data using the Performance Monitor. Be sure to collect this data during peak activity for at least a one-hour time period using 10 minute sample intervals.
2. Print the Performance Tools System Report using the PRTSYSRPT command. Refer to the ″Disk Utilization″ section of the Performance Tools LPP System Report. From this report, the following data is obtained:
   - Total operations per second - use Op Per Second column.
   - Total GBs of DASD installed - use Size (M) column.
3. To determine the total GBs installed, simply add the ″Size (M)″ column and divide by 1000. When adding the total GBs, you should **only** include the disk units you plan to replace. Also, if Mirroring is active, divide the total GB being mirrored by two when calculating the sum.

4. To determine the total operations per second, add the total operations per second number (″Op Per Second″ column). When adding the total operations per second, you should **only** include the disk units you plan to replace. Also, if mirroring is active, you need to divide the total number of operations per second for all mirrored units by two.
5. To determine the operations per second per GB, divide the total operations per second you calculated in Step 4 by the total GBs installed value you calculated in Step 3.

You can use the operations per second value to determine what model of DASD best fits your current or projected DASD performance requirement.

5.3 DASD Performance - Commercial Batch (Base versus RAID)

This section shows the results of running an IBM batch workload in a dedicated environment. The workload performs the following functions:

- Sequential and Keyed Record Copy
- Sequential and Keyed Program Read
- Sequential and Keyed Record Read/Update
- Record Matching
- Adding and Removing Members
- RGZPFM of 500,000 Records
- Average - 40% Read Ops, 60% Write Ops, 17 KB/IO
- 70% Synchronous / 30% Asynchronous Ops

The workload was run in a 24MB memory pool. Various DASD subsystem configurations were tested, operating in BASE mode and High Availability (HA or RAID-5) mode.

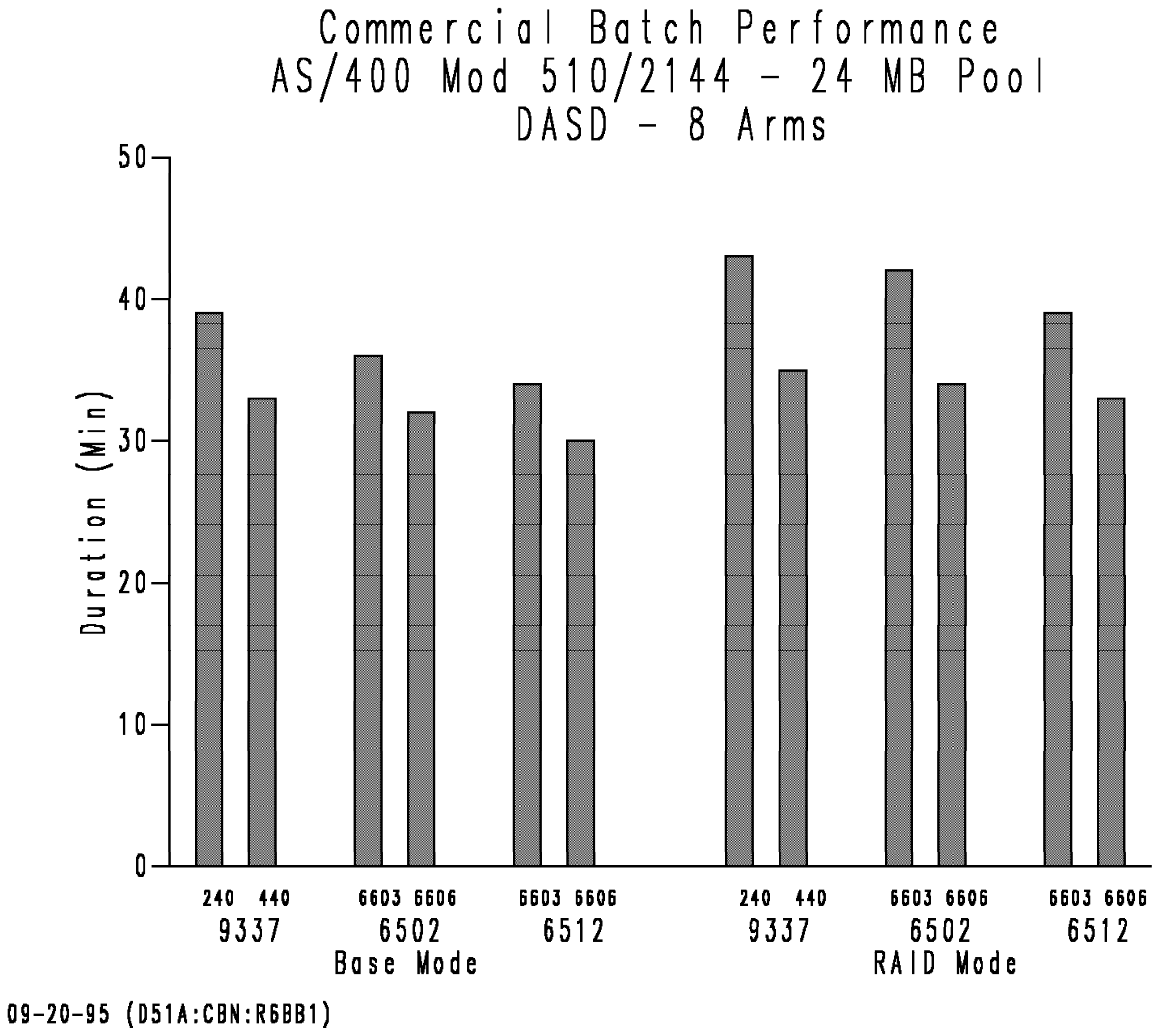


Figure 18. Commercial Batch Performance - Base versus RAID

The difference in batch run time for this workload did not vary greatly between BASE operation and HA operation.

Note: This workload is not typical of any particular customer environment, and should be used as guide only. The results obtained in a customer environment are different, and a capacity planning tool, such as BEST/1, should be used.

5.4 DASD Performance - General

Mixing RAID DASD with other DASD in one ASP: Combining 6502/6512 RAID DASD with mirrored DASD in a single ASP is allowed. Combining RAID DASD with mirrored DASD on the same 6502/6512 is also allowed.

Write Intensive Applications (for example, RESTORE): RAID-5 (such as system checksum) can have a significant impact on batch type programs that issue many writes in a short period of time. This is due to the four times increase in disk IO required for each write. The 6502/6512 write cache handles this impact for almost all scenarios except those that write hundreds of writes to the DASD in a short period of time. Even in this worst case scenario with only one

6502/6512 array configured, restoring a large file took only 30% longer than restoring to a ″standard″ 9336-0x0 or 9337-0x0 configuration.

The 6502/6512 can restore small objects faster than 9336-0x0 and 9337-0xx because of the non-volatile write cache (2MB for a 6502 and 4MB for a 6512). The write cache provides fast completion of write requests and is able to ″stay ahead″ of the system.

The 6502/6512 RAID models offer significant advantages in availability, reliability, price, and so on. One of the ″costs″ of the availability advantage is the increased time to restore data. This increase in time needs to be considered when planning for the installation of a RAID-5 disk subsystem. The time to load data onto a RAID-5 disk subsystem must be included in the overall installation planning. With the increased availability and reliability offered by RAID DASD when attached to a 6502 or 6512 IOP, the necessity to reload the data again due to a single disk unit failure is virtually eliminated.

DST ″Add Unit″: Part of the process of adding DASD to a system is using the Dedicated Service Tool (DST) to ″Add Unit″. This ensures that the entire DASD (or DASDs) is initialized with a X′00′ data pattern and verified.

When multiple DASD are added at once, the system adds up to 16 units in parallel.

The time for adding up to 16 units on the 6530, 6502, or 6512 (base mode) is approximately:

- 24 minutes for 1GB arms (6602)
- 48 minutes for 2GB arms (6603/6606)
- 86 minutes for 4GB arms (6607)

The Dedicated Service Tool is also used to start and stop parity (RAID-5) arrays on the 6502 or 6512. When a parity array is initially set up, the fastest approach is to start parity on an array first and then add the arms to an ASP. The time required for this process (start parity and add) on two eight-arm arrays is approximately:

- 24 minutes for 1GB arms (6602)
- 48 minutes for 2GB arms (6603/6606)
- 86 minutes for 4GB arms (6607)

If the arms are added to the ASP before starting the array, the time required may double. If a system IPL occurs between starting the array and adding the arms to an ASP, the time required can be three times as long.

If the arms are currently part of an ASP, starting an array takes longer because the system may need to move data before it synchronizes the parity stripes. This can take up to:

- 45 minutes for 1GB arms (6602)
- 90 minutes for 2GB arms (6603/6606)
- 160 minutes for 4GB arms (6607)

Stopping parity on an eight-arm array takes about:

- 40 seconds for 1GB arms (6602)
- 70 seconds for 2GB arms (6603/6606)
- 120 seconds for 4GB arms (6607)

Determining Current DASD Subsystem Performance: In order to determine your current DASD subsystem performance requirements, you need to calculate the Ops/Sec/GB value for your current workload using the Performance Monitor. Use the following steps to get representative data for DASD activity:

1. First, determine when to collect performance data in order to best characterize how the system is using DASD. Typically, the system should be monitored during periods of peak activity. Data should be collected over at least a one-hour period using 10-minute sample intervals. At least three of these snapshots should be gathered in order to get a representative sample of DASD activity.

2. Collect performance data using the Start Performance Monitor STRPFRMON INTERVAL(10) HOUR(1) command. Use the defaults for other command parameters. For additional information on collecting performance data, refer to *OS/400 Work Management Version*, SC41-4306, or to *Performance Tools/400 Version 3*, SC41-4340, for more information.

3. If Performance Tools/400 is installed, print the system report for the measurement using either the Print System Report (PRTSYSRPT) command or using the performance tools menu interface (GO PERFORM). If the Performance Tools/400 is not installed, save the performance data to tape and restore it to a system where Performance Tools/400 has been installed (at the same release level or a later release). If the release is at a later level, it may be necessary to convert the performance data using the Convert Performance Data (CVTPFRDTA) command.

After collecting the performance data needed to characterize the current DASD subsystem performance, continue with the instructions that follow Figure 19 on page 85 to calculate the Ops/Sec/GB for the installed DASD. The Disk Utilization report is provided as an aid in following the general instructions.

```
                                                          System Report                                   09/12/96 13:36:10
                                                         Disk Utilization                                        Page 0001
                                                  Data collected 091096 at 1200
Member  . . . : Q962541200 Model/Serial  . : 510-2144/10-11111   Main storage . . :  384.0 M   Started  . . . . : 09/10/96 12:01:45
  Library . . : QPFRDATA   System name . . :         GENSYS02   Version/Release  :   3/ 7.0   Stopped  . . . . : 09/10/96 13:56:43
      Unit                  Size    IOP   IOP           ASP     --Percent--     Op Per    K Per    ---- Average Time Per I/O ---
Unit  Name         Type    (M)     Util  Name          ID      Full   Util     Second      I/O    Service     Wait     Response
----  ----------  ----  -------   ----  ------------  ---     ----   ----   ---------  --------- -------     ------     --------
0001  DD002       6606    1,967    3.6  CMB01          01     93.1    4.8        4.78      14.5   .0100      .0031       .0131
0002  DD003       6606    1,967    3.6  CMB01          01     89.5    9.6        8.59       9.5   .0111      .0018       .0129
0003  DD001       6105      320    3.6  CMB01          01     92.8    2.4        1.17       8.1   .0203      .0051       .0254
0004  DD004       6105      320    1.4  SI01           01     92.9    1.9         .94       9.0   .0200      .0032       .0232
0005  DD005       6105      320    1.4  SI01           01     93.0    2.1        1.03       8.4   .0203      .0048       .0251
0006  DD006       6105      320    1.4  SI01           01     93.1    1.9        1.01       8.5   .0188      .0069       .0257
0007  DD009       9337      970   13.3  SI03           01     89.6   13.6        7.03       7.5   .0193      .0041       .0234
0008  DD008       9337      970   13.3  SI03           01     89.6   14.1        7.32       6.8   .0192      .0038       .0230
0009  DD007       9337      970   13.3  SI03           01     89.6   12.7        6.44       7.6   .0197      .0031       .0228
0010  DD010       9337      970   13.3  SI03           01     89.6   12.8        6.65       7.7   .0192      .0033       .0225
0011  DD011       9337      970   13.3  SI03           01     89.6   12.5        6.44       7.4   .0193      .0036       .0229
0012  DD012       9337      970   13.3  SI03           01     89.6   12.0        6.55       7.2   .0182      .0040       .0222
0013  DD013       9337      970   13.3  SI03           01     89.6   12.5        6.53       7.6   .0191      .0038       .0229
0014  DD023       9336      857    5.6  SI04           01     89.6    4.0        2.56      12.1   .0155      .0031       .0186
0015  DD015       9336      857    5.6  SI04           01     92.0    8.3        5.93       5.7   .0139      .0014       .0153
0016  DD016       9336      857    5.6  SI04           01     89.6   11.3        6.72       6.5   .0168      .0023       .0191
Total                    14,575  1
Average                                                       90.5    8.5        4.98       8.0   .0170      .0032       .0202
******************************************************************************************
*  TOTAL I/O's per second (you must total this manually)                       79.69  2  *
******************************************************************************************

Unit                   --  Disk arm identifier
Unit Name              --  Disk arm resource name
Type                   --  Type of disk
Size (M)               --  Disk space capacity in millions of bytes
IOP Util               --  Percentage of utilization for each Input/Output Processor
IOP Name               --  Input/Output Processor resource name
ASP ID                 --  Auxiliary Storage Pool ID
Percent Full           --  Percentage of disk space capacity in use
Percent Util           --  Average disk operation utilization (busy)
Op per Second          --  Average number of disk operations per second
K Per I/O              --  Average number of kilobytes (1024) transferred per disk operation
Average Service Time   --  Average disk service time per I/O operation
Average Wait Time      --  Average disk wait time per I/O operation
Average Response Time  --  Average disk response time per I/O operation
```

Figure 19. Performance Tools System Report, Disk Utilization

1. Refer to the ″Disk Utilization″ section of the System Report. The following data can be obtained from this report:

   - Total operations per second - use Op Per Second column.
   - Total GBs of DASD installed - use Size (M) column.

2. To determine the total GBs installed, simply add the ″Size (M)″ column and divide by 1000. When adding the total GBs, **you should only include the disks you plan to replace.** Also, if mirroring is active, divide the total GB being mirrored by two when calculating the sum. In the example ″Disk Utilization″ report, the total number of GBs is 14.575GB. 1

3. To determine the total operations per second, add the total operations per second number (″Op Per Second″ column). When adding the total operations per second, you should **only include the disk units you plan to replace.** Also, if mirroring is active, you need to divide the total number of operations per second for all mirrored units by two.

   In the example ″Disk Utilization″ report, the total operations per second is 79.69. 2

4. To determine the operations per second per GB, divide the total operations per second you calculated in Step 3 by the total GBs installed value you calculated in Step 2.

   The Ops/Sec/GB in this example is 5.5 Ops/Sec/GB.

Select the DASD Model that Meets Your Needs: You can use the Ops/Sec/GB value to determine what model of DASD best fits your current or projected DASD subsystem performance requirement. Use the charts in Section 5.2.5, "Ops/Sec/GB Guidelines for DASD Subsystems" on page 78 to assist in choosing the appropriate internal or external DASD.

Use the topmost value of Ops/Sec/GB bar for each disk model shown as a guide for its disk I/O limit for acceptable performance and to allow for future growth in disk I/O capacity. That is, if your current disk Ops/Sec/GB is 18, do not select a new disk model with a topmost bar value in the 18 to 20 range.

Chapter 6. Performance Management Methodology

This chapter presents the methodology for performance management. A flowchart is used to describe the organization of information in Chapter 6, “Performance Management Methodology” through :hdref=ch09 page=no. in this book. It should be used to develop a procedure for establishing performance objectives or guidelines, setting up a plan for collecting and evaluating that performance data using the Performance Tools licensed program, and other AS/400 performance tools.

This flowchart separates performance management into three major stages:

- Performance measurement
- Performance trend analysis
- Performance problem analysis

Each stage has several ″activities″ that are discussed in this publication. Note that if you already have a formal performance management methodology, you may consider integrating portions of the methodology presented herein with your existing procedures.

A compressed image of the following flowchart appears at the start of Chapter 6, “Performance Management Methodology” through Chapter 8, “Performance Trend Analysis.” This is to make it easier to reference information presented in each of those chapters to procedures discussed here for the performance management flowchart.

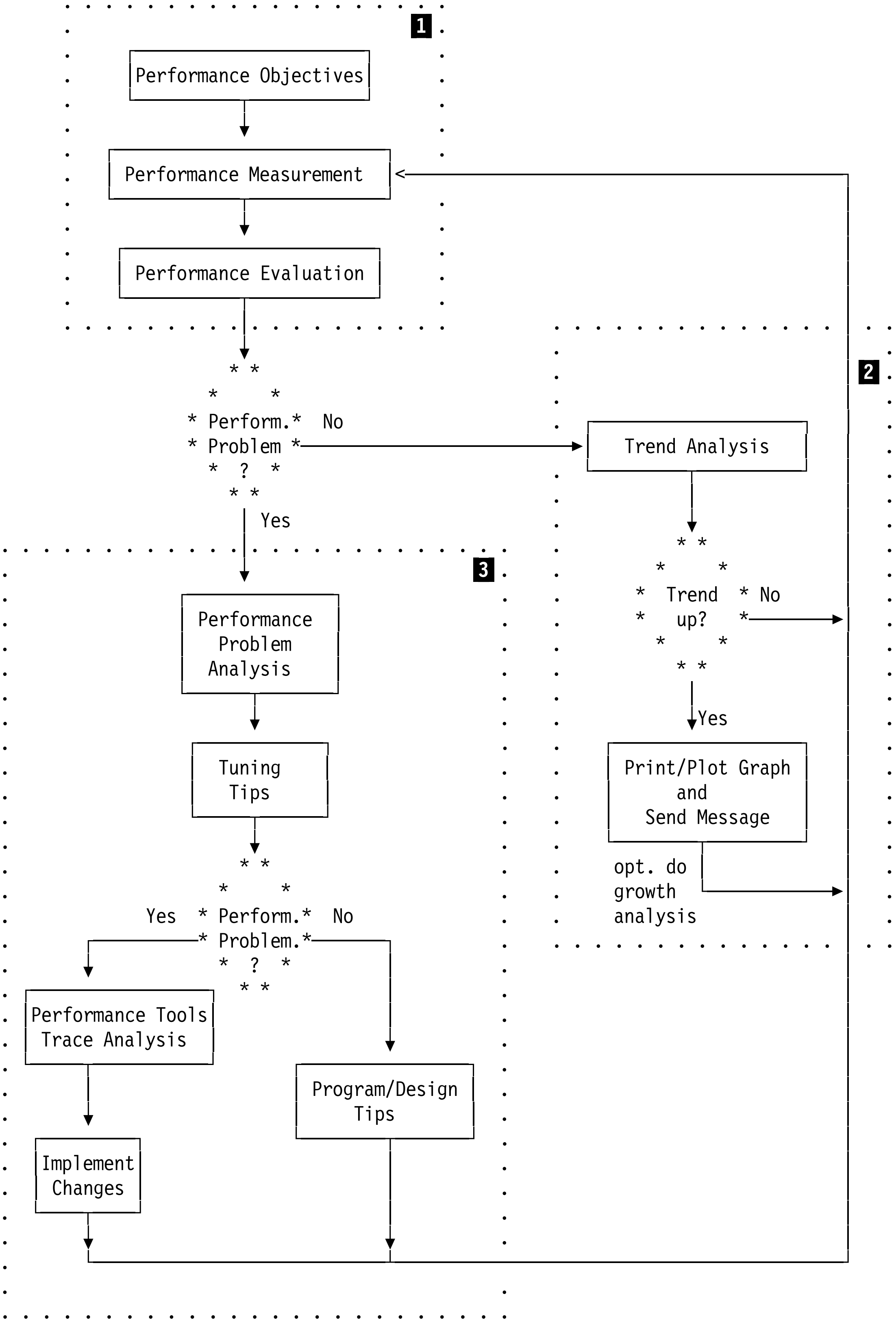


Figure 20. Performance Management Flowchart

6.1 Performance Management and Review

Stage **1** of the performance management flowchart is discussed in the following paragraphs. Performance measurement requires evaluating collected performance data against a set of defined objectives. Chapter 9, "Performance Problem Analysis" provides additional details.

Performance Objectives: This activity addresses the definition of customer objectives or expectations in a manner that can be measured. This includes ″specifications″ for batch and interactive work such as response time for local and remote workstations, the definition of an interactive transaction, and the expected transactions per hour. The definition of batch job throughput is based on a time period and records processed or lines of printed output during that time period. In a communications environment, the objective is the number of records transferred to another system in a specified time period.

The customer should be aware of what can and cannot be expected from the current configuration. If a purchased application package is used, performance information for a similar configuration should be reviewed. For example, performance of an application with four active workstations should not be used to indicate performance in a 50-active workstation environment.

It is important that the customer define in detail the objectives of performance that is expected for each type of job (for example, response time for local and remote terminals, throughput for batch jobs, office tasks per hour, and so on).

Performance Measurement: This step covers the methodology for planning and executing performance measurements. Performance measurement criteria should be adopted in order to perform a periodic run of the monitor. This criteria should consider such things as day and time to do the collection, duration, type of information to be collected, and so on. The customer should define a peak period (or critical applications) to be measured.

Performance data can be collected from both running the system performance monitor and observations made by the system operator or end users at their workstation. Using both sets of information is a good way to evaluate system and application performance.

Observations made by operators or end users should include date and time of day, user ID, application function, or display function being performed and other information that may be unique to a particular application environment. If a problem occurs, any unusual considerations should have been documented. However, this kind of observation may not give detailed information of all components or jobs that might be needed for problem analysis.

It is also important to collect performance data and review it even when there have been no reported performance problems so that potential problems may be detected before they impact the daily production environment.

Performance Evaluation: Often the indication of a performance problem does not occur until one or more workstation operators (end users) reports a problem, or a system operator indicates some batch jobs are taking longer to complete. Having a procedure for collecting performance data is important. However, performance management stresses collecting and evaluating OS/400 performance monitor data to detect potential problems before they are apparent to end users.

Appendix A, “Guidelines for Interpreting Performance Data” contains guidelines for system resource usage that should be compared to information in the Performance Tools Reports. This chapter provides guidance on how to use the various reports.

Performance Problem Action: If there are no performance problems reported by the end users and the operator has not seen any abnormal situation, the collected information should still be stored and reviewed for further trend analysis. If problems are reported by users or trend analysis indicates good performance guidelines have been exceeded, performance problem analysis must be performed. Detailed performance data analysis is required if the basic tuning tips do not resolve the problem.

Trend analysis requires evaluating ″past and current″ sets of performance data. In this document, a set of objects and programs are provided to automate the performance data analysis to detect trends. Within this document, this automated tool is referred to as the ″Automated Performance Management Tool″. Further details on this tool are provided in Appendix C, “IBM Internal Use Only Tools/Documents.”

6.2 Performance Trend Analysis

Stage **2** of the performance management flowchart is discussed in the following paragraphs. Chapter 8, “Performance Trend Analysis” provides additional details.

Trend analysis involves the review of performance measurement data collected over a period of time and looks for relationships or ratios between interactive throughput, response time, and the utilization of the various system resources that impact the level of performance you can expect.

By doing this, you should be able to see the pattern of your system workload, know whether it is steady, increasing or decreasing, and know what percentage utilization you are using on your system.

Trends in performance or machine utilization from month to month can be easily spotted by issuing the Display Historical Graph (DSPHSTGPH) command. Historical data can be created using the CRTHSTDTA command or by the Display Historical Data with option *YES specified on the Create Historical Data field.

Performance trend analysis is done automatically once every month within the Automated Performance Management Tool by detecting a difference between system value QMONTH and the value stored in data area MONTH in library PFRMGMT. If a difference is detected, an RPG program is called that checks if one of the resources or the response time is showing an upward trend. It does this by comparing the collected data from the last month with the data from all previous months.

If the resource utilization is trending upwards or performance is degrading, appropriate information should be made available to the installation management.

What to Do When the Trend Exceeds Certain Thresholds: Take an ″above average″ measurement and extrapolate with the measured growth rate into the future using the Capacity Planning support (BEST/1) under the Performance

Tools licensed program. The measured data selected for capacity planning should be one showing a medium-to-heavy workload that the customer feels accurately reflects the ″typical workload″ they want to use for capacity planning. Typically, this does *not* include ″once-a-year″ workloads, such as running five long-running queries at the same time or some other excessive CPU utilization or high disk I/O jobs.

Capacity planning involves predicting when you are going to run out of capacity on your present system, knowing by what percent you are likely to grow over a given period of time, and projecting what hardware upgrades are required to adequately handle that growth.

This ″art″ is much more accurate when using measured data from performance data monitor (STRPFRMON) or automatic data collection (ADDPFRCOL) under the Performance Tools licensed program.

6.3 Performance Problem Analysis

Stage **3** of the performance management flowchart is discussed in the following paragraphs. Chapter 9, "Performance Problem Analysis" provides additional details.

If a problem is reported, a step-by-step procedure should be followed in order to collect the information necessary to resolve the problem.

This procedure should include the utilization of OS/400 storage pool size and activity level adjustment support available through the system value QPFRADJ, the Work with System Status (WRKSYSSTS) command, and the Performance Tools Advisor function (ANZPFRDTA) command or menu options.

In most situations, the Advisor output, the Print Summary (PRTSYSRPT) command, and Component (PRTCPTRPT) command reports should be the first set of Performance Tools output reviewed.

Review the Performance Tools Advisor output first as it is a ″fast path″ through performance monitor *sample data* for the Advisor using the Performance Tools/400 (5716-PT1) licensed program (Advisor is available with Manager or Agent feature), or *sample data* and *interactive trace data* for the Advisor under the Version 3 Release 6 Performance Tools/400 (5716-PT1) licensed program. The Advisor function provides recommendations and conclusions, and includes analysis of communications line activity, but does not provide important response time information contained in the Summary and Component reports. The system report provides system-wide information in determining whether overall customer performance objectives are being reached.

If the system is busy or is already experiencing poor response time, it is preferable to run the Advisor command (ANZPFRDTA) or functions during a period of low system activity or at least with a lower priority than currently active interactive jobs. Either use the CHGJOB command or the Submit Job (SBMJOB) command to run ANZPFRDTA as a batch job. ANZPFRDTA is a CPU-intensive process.

WRKSYSACT (Work with System Activity) is a less CPU-intensive command than ANZPFRDTA command or WRKACTJOB command, and can be used interactively to identify system and user jobs consuming significant resources that may assist in identifying functions causing poor performance. On large AS/400 systems, if you are using WRKSYSACT with the refresh option, set it to 20 seconds or higher to reduce the resources consumed by this command.

Appendix A, “Guidelines for Interpreting Performance Data” contains performance-related guidelines that include threshold figures for some key values, such as interactive CPU utilization, line utilization, and so on. These values should be compared with those obtained from the Performance Tool reports. Any discrepancy indicates a potential problem that should be analyzed in more detail.

In addition to the Performance Tools Work with System Activity (WRKSYSACT) command, consider the Display Performance Data (DSPPFRDTA) command and the reports produced by the Print Transaction (PRTTNSRPT) command.

PRTTNSRPT requires that TRACE(*ALL) be specified on the Start Performance Monitor (STRPFRMON) command. The output of this command can be used to evaluate details for specific jobs that may be encountering performance problems.

Based on the analysis of the data provided by the previous display and printed output, you may determine that using one or more of the following performance tools functions is required for more detailed analysis:

- Trace Job commands such as STRJOBTRC, STRSRVJOB, TRCJOB, ENDJOBTRC, ENDSRVJOB, and PRTJOBTRC. Job trace output may identify which user programs or IBM programs and modules are called, and which of these programs are consuming large amounts of CPU or causing a large number of page faults.
- Using the AS/400 Performance Tools Performance Explorer (PEX) STATS option functions to list detailed CPU usage and disk accesses by programs, Licensed Internal Code (LIC) tasks, and modules. This function replaces the pre-V3R6 TPST PRPQ.
- Using the AS/400 Performance Tools Performance Explorer (PEX) PROFILE option functions to identify high CPU processing time instructions within a program. This function replaces the pre-V3R6 SAM functions.
- Print object lock support (PRTLCKRPT) command for further refinement of object seize/lock holders and waiters.
- Analyze database file and program relationships and usage (ANZACCGRP, ANZDBF, ANZDBFKEY, and ANZPGM commands).
- Using the AS/400 Performance Tools Performance Explorer (PEX) TRACE option functions to identify jobs, programs, objects, storage pool, and disk device activity. This option replaces the pre-V3R6 SMTRACE and STRDSKCOL/ENDDSKCOL commands.

Tuning Tips: Tuning is a way of adjusting a system′s performance to meet the requirement. Customers must clearly understand their performance objectives. At this point, the flowchart shows what to do if there is performance degradation in the systems.

For basic tuning, automatic systems tuning is a useful method to maintain good performance. This can be done by setting the system value QPFRADJ to indicate that system tuning adjustments are to be performed at IPL time or dynamically while the system is running.

Do not use the system QPFRADJ and SETOBJACC at the same time for a shared pool. QPFRADJ removes storage from a shared pool that has no paging activity. If SETOBJACC is used to pre-load an object into the same pool, it loses some of its storage. When SETOBJACC is used properly, there is no page faulting in the pool and, therefore, QPFRADJ considers that pool a prime candidate for removing storage.

Alternatively, consider the Performance Tools/400 Advisor as a tool for reassigning memory across storage pools. It recommends storage pool reassignment and makes the change if requested by operator response. The Advisor analyzes performance data that is collected by the Performance Monitor and produces recommendations and conclusions to improve system performance.

The Advisor can perform changes to pool sizes and activity levels as it recommends in the Advisor analysis. These changes are not made dynamically, but only after the operator tells the Advisor to make the changes. The changes can be made to all shared pools on the system, including the machine pool, but not to user-defined pools.

The Advisor can also analyze performance data collected from other AS/400 systems, but you cannot have the Advisor change another system since the recommendations are not for the same machine.

The Advisor does not make any changes if the pool configuration has changed since the performance data was collected, or if the automatic performance adjustment facility was turned on using QPFRADJ values.

The Advisor does not identify or fix all performance problems, or produce recommendations for modifying specific application programs to improve their performance, but in many cases, the Advisor might be the only tool required to make the improvements you need.

Chapter 11, “System Performance Tuning Tips” in this publication provides more detailed information for tuning tips and technique for those methods and other specialized tools.

Performance Tools Trace Analysis: After system tuning, new performance measurements should be taken. If the problem has not been resolved, additional analysis must be performed by collecting the performance data with the STRPFRMON command option TRACE(*YES). This option collects data that enables you to see which jobs, programs, or transactions are causing the heaviest resource utilizations. This is where the application and, if necessary, the program tuning analysis process begins. Performance Tools output can separate application programs from system programs or tasks.

Since V2R2, the Advisor processes STRPFRMON trace data for interactive jobs. You should utilize this tool first before using other ″specialized″ tools.

Trace data is used for the Transaction report and Lock report output. Transaction report data includes much more detail per job regarding sign on and sign off, response time, CPU seconds, disk I/O operations, object lock/seize times, time slice end, short wait times, and so on.

Tip:

If the Transaction report shows seize/lock conditions, the Print Lock Report *TOD option should be reviewed. Use a time limit value of 50 milliseconds (filter all seize/lock conflicts less than this value) and use the Time of Day option to bracket the reporting based on when the other trace data showed the problem to occur. This reduces the amount of ouput you have to look at and probably still contains relevant data.

Consider using the Performance Tools trace job support to identify programs taking excessive amounts of CPU during an interactive transaction. Do not produce the detailed job analysis until you identify programs or jobs you want to analyze. The summary reports allow you to determine the overall performance of the job without analyzing the Detail report. They provide a summary of the number of program initializations and full database file opens and closes as well as other information. These two items are often the cause of a large number of performance problems. There is another tool called PRTTRCSUM in IBMLIB that produces an excellent view of the program-to-program call structure using the Trace Job data.

The AS/400 Performance Tools Performance Explorer (PEX) *STATS option may be used to identify which programs to investigate. PEX *STATS output shows which programs, processes, tasks, and modules are using large amounts of CPU time or numbers of disk accesses. The redbook *AS/400 Performance Explorer: Tips and Techniques*, SG24-4781 provides more information on PEX *STATS.

Change Implementation: If you decide to change system or configuration parameters, the use of system commands (for example, run in batch rather than interactively), or your application, you need to identify and rank the changes. Performance benefit versus the cost to change are key because they could be unique to each customer environment.

When changes are made, they should be made in a controlled environment and measured. Consider the following when implementing changes:

1. Make only one change at a time.
2. Take several measurements before reaching a conclusion.
3. Measurements must be made under similar conditions; that is, the same workload, number of active workstations, number of batch jobs, and so on must be valid for purposes of comparison.
4. Performance analysis is a long process. It is much easier to integrate performance tips as discussed in Chapter 12, “Design and Coding Tips” during application development than it is to add them to existing applications.
5. A significant improvement is most commonly the result of several small changes.

Application Design Tips: If a problem appears to point to excessive CPU utilization by one or more programs, the PEX PROFILE support should be used to identify important program instructions. PEX PROFILE output shows the relative amount of processing time spent in different parts of a program or a set of programs.

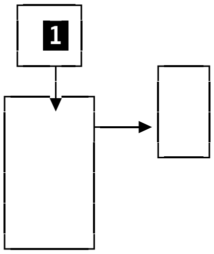
Chapter 12, “Design and Coding Tips” of this publication has a number of application design tips for programming that can assist in maximizing performance.

In some cases, the tips listed can be used to improve the performance of existing applications by merely making a few small changes. However, in many cases, integrating the tips into existing applications can be time consuming and exposes the application to regression problems. In those cases, adding hardware or including the tips in new applications may be the best alternative.

Note: Understanding these tips can assist in the evaluation process when choosing an existing application package that satisfies customer functional requirements but also must meet customer performance objectives.

Chapter 7. Performance Management and Review

This chapter discusses stage 1 of the performance management flowchart.



7.1.1 Major Topics Covered in This Chapter

- How to set your performance objectives.
- A suggested method of collecting performance data manually or with an automated process.
- A suggested method of how to summarize and store this data with minimal disk space allocation.
- A description of the methodology and programs involved in the automated performance data collection and initial analysis process.

Chapter 8, “Performance Trend Analysis” and hdref refid=ch10 page=no. are written with the assumption of using this data measurement technique. The readers of this document are free to supply their own modifications to this measurement process and the trend and problem analysis procedures discussed in Chapter 8, “Performance Trend Analysis” and Chapter 9, “Performance Problem Analysis”

7.2 Develop a Measurement Plan

The performance management cycle begins with setting the performance objectives for your computer installation.

Perhaps the best way to start is to estimate the maximum hourly and daily interactive transaction throughput required of your computer system during your peak business periods. After that, you can decide what average response time is acceptable to your local and remote workstations.

You should think about how long your regular batch processes take, and how to schedule them so that they complete in time to achieve your business requirements. If no application performance data exists, you can use QSIZE400 to establish some performance objectives. The Performance Tools Capacity Planning function (BEST/1) can also be used if you have application details to use as input to workload and objectives displays in BEST/1.

If the application is already available, Performance Monitor output can be used to initially help establish a base set of statistics that should be documented in a performance objective plan containing:

- The peak transactions per hour
- The peak transactions per day
- Acceptable average response time for local workstations
- Acceptable average response time for remote workstations:

  The transaction response time should be specified separately because remote workstations are almost always one to two seconds slower than local workstations. Expected response for simple and complex transactions can also be specified separately if required.
- A list of the major scheduled batch jobs with times when they are run and their expected duration
- A list of other unscheduled batch jobs that may be required

7.3 Technique for Measurement

Measuring the performance of your computer system on a **regular basis** and finding a way to **summarize and save this information** is one of the most important parts of performance management. If done properly, it gives you these advantages:

- It provides a clear indication of your current workload and the percentage growth of this workload.
- It shows the variation in this workload over a daily, weekly, or monthly time frame.
- It shows the effect of operational changes, tuning adjustments, and application design changes.
- It allows you to find out what percentage of your computer system′s capacity you are currently using, and should help you to know when you are going to run out of capacity.
- It should help you budget for upgrades well in advance and minimize the risk of a performance crisis.

7.3.1 Suggested Method of Using the Performance Monitor

The following sections describe how you can manually collect performance data, produce some performance reports, summarize the data, and list the summary performance data. You can delete old performance data members once you have no further need for them, yet retain the summarized performance information for later use.

Experience has shown that the more manual this process is, the less likely it is to be followed on a regular basis. We recommend that you use the automated procedure and programs for performance management because it is unlikely that you will continue using this process and derive the full benefit from doing so unless it is automated.

A question sometimes asked is ″how often and for how long should one run the Performance Monitor on an AS/400 system?″

The answer to this is usually only for one to two hours during a period when your system is busy because your system should be measured and judged by its ability to perform well at peak workload.

The following information suggests how you can use the AS/400 Performance Monitor under various circumstances:

To Measure Total Daily Throughput or Workload Variation

Run the performance tools (STRPFRMON) for the entire shift with default collection intervals of 15 minutes and with ″sample data″ collection only. Use this data to establish a record of system performance and to understand the characteristics of the system′s performance.

You should collect sample data regularly so you can compare the collected data within the similar workloads.

Regular Performance Monitoring

Run the Performance Monitor for periods of one to two hours with five minute sample intervals at the same time each day or at least two to three times a week. Collect sample data only and pick a time when your system is busy.

You can also ask the users which period of time they think the system has a performance problem. Run Performance Monitor during that period.

Permit the STRPFRMON command to generate a default member name, which gives you a *date/time stamp* in the name. This is important when the results are summarized and collected.

For Detailed Performance Analysis or when a Performance Problem Exists

Run the Performance Monitor for one to two hours at a busy time, *but collect trace data as well as sample data.* Collecting trace data places a slight additional load on your system, but it is necessary to get the additional detailed information about transactions, jobs, programs and locks, and so on. Dumping the trace data itself can be a three to five percent increase in CPU utilization, so dumping the trace data during a non-busy time period should be considered. End the performance monitor with ENDPFRMON DMPTRC(*NO) and use the DMPTRC command at a later time.

On a fluctuating workload situation, run Performance Monitor also during the slope (lowest) workload before and after peak hours to find out what really happens during those times.

On a system with high transaction rates, or with a large number of jobs, the trace table size (allocated by the system) may not be large enough to contain more than one hour of data. Once the trace table has been filled, some performance report data may be incomplete. You can increase the trace table size with the STRSST command. The Transaction Report - Job Summary report indicates the trace time period under the System Summary Data heading.

7.3.2 Automatic Data Collection

Automatic data collection allows you to select specific days of the week to collect data using the OS/400 performance monitor. Using the ADDPFRCOL command or option 1 (add) on the Work with Performance Collection (WRKPFRCOL) menu lets you establish a regular schedule for collecting performance data automatically on any day of the week.

You must specify the day and time to collect performance data, or just specify starting and ending times and run it every day of the week. Make sure the collection time includes the peak hours or period that you want to monitor.

7.4 Performance Review Options

There are a number of tools that can be used to review performance data. Some can be used on an ad hoc basis, while others are more sophisticated and automatically monitor for performance trends. This section reviews a subset of these tools.

7.4.1 Advisor

The Advisor provides the easiest way to evaluate performance data. It fits into the set of performance tools between automatic system tuning and other specialized tools or performance reports. Option 10 on the AS/400 Performance Tools Menu or the ANZPFRDTA command leads you to the Recommendations and Conclusions Display of AS/400 performance.

The Advisor analyzes performance data you collect with the Performance Monitor. You can use Advisor to analyze performance data restored from other systems.

The Advisor analyzes performance data including:

- Storage pool size
- Activity level
- Disk and CPU utilization
- Communications utilization and error rates
- Input/output processor utilization
- Unusual job activities-exceptions or excessive use of system resources
- Interactive trace data (when collected)

The Advisor does not:

- Make any recommendations for modifying specific programs to improve their performance.
- Analyze noninteractive trace data.

The Advisor analyzes one member set of performance data at a time. You need to select the member that was collected when the performance problem occurred. It is easy to find the right time interval to analyze with the Display Histogram function. For instance, if you need a time interval when the transaction has a longer response time, select the Transaction Response Time option on the Display Histogram display. From the chart, you can select a time interval by moving the cursor to that interval, typing a ″1″ and pressing the Enter key. The Advisor analyzes performance data at that particular interval and gives you recommendations and conclusions.

Beginning with Version 2 Release 2, the default Advisor action is to analyze interactive job trace data. To avoid analyzing trace data, use the ANZPFRDTA command, press F4 (prompt), and F10 (additional parameter) to change the value of the DATATYPE parameter to *SAMPLE.

The recommendations suggest changes to the basic system′s tuning values that can improve performance. They also list problems that can solve other performance problems. You can get more detailed suggestions by typing a ″5″ in the Option column.

The Conclusions display lists conditions that can affect performance when analyzed data is collected. These can include thresholds reached, save and restore activities, teleprocessing line errors, and some other factors.

You can use the conclusions that are not related to recommendations as guides for collecting more performance data or for adjusting the system. Type a ″5″ in the Option column to see more detail about a particular conclusion.

You can refer to Section 9.5.11, “Advisor” on page 139 for more information about Advisor.

7.4.2 Performance Graphics

Performance data collected by the Performance Monitor can be displayed in graphical format using the Performance Graphics option in Performance Tools/400. The graphs can be displayed interactively, printed, or plotted to hardcopy. You can use the IBM-supplied format (in QPFRDATA) or create your own format.

Performance Graphics uses the performance data collected from a single run of Performance Monitor. You can select to display a graph from a specific member that has jobs with poor performance.

Performance Graphics can also be used to show historical data. Historical Data graphs use several runs of Performance Monitor. Use this graph to show the trend of performance, or how the performance of the system has changed over time. The period of time when a performance problem occurred can easily be spotted in an Historical Data graph.

7.4.3 Performance Management/400

Performance Management/400 (PM/400) is an IBM system management service offering that assists customers by helping them plan and manage system resources through ongoing analysis of key performance indicators.

The service uses a set of software and procedures installed on the customer′s system. The software collects performance data, summarizes, and transmits the summarized data to the AS/400 Competency Center in Rochester, Minnesota weekly.

PM/400 automates these functions and provides a summary of capacity and performance information. Reports and graphs are produced in a format that both non-technical and technical persons can understand.

Performance data is analyzed and maintained by IBM. Section 10.2, “Performance Management/400 (PM/400)” on page 147 provides more information on PM/400.

PM/400 does not require AS/400 Performance Tools (5716-PT1) and has no intention to replace that product.

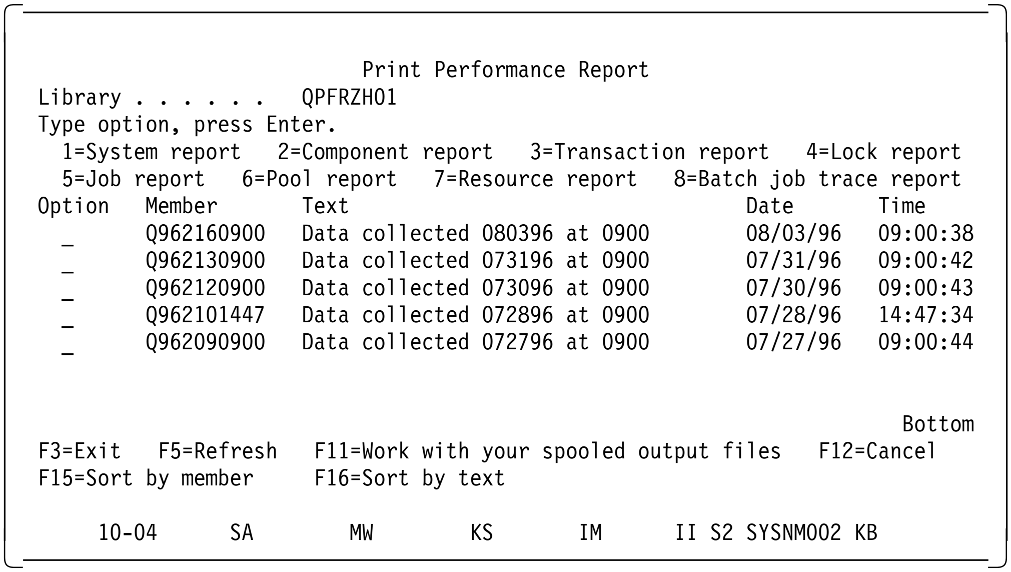
7.4.4 Performance Tools Report

Printing performance reports provides complete information on collected performance data. You can review the performance of specific jobs, transactions, or other performance elements. This can be done by taking Option 3 (Print Performance Report) on the AS/400 Performance Tools menu which gives you the Print Performance Report display, or by issuing the following commands from any command line.

- Print System Report (PRTSYSRPT)
- Print Component Report (PRTCPTRPT)
- Print Transaction Report (PRTTNSRPT)
- Print Lock Report (PRTLCKRPT)
- Print Job Report (PRTJOBRPT)
- Print Pool Report (PRTPOLRPT)
- Print Resource Report (PRTRSCRPT)
- Print Batch Job Trace Report (PRTTRCRPT)

Each of these commands gives information with a different level of detail. The System report, Component report, Job report, Pool report, and Resource report are produced from sample data collected through the STRPFRMON command. If you collect trace data using the STRPFRMON command, you can produce a Transaction report, Lock report, or Batch Job Trace report from this information.

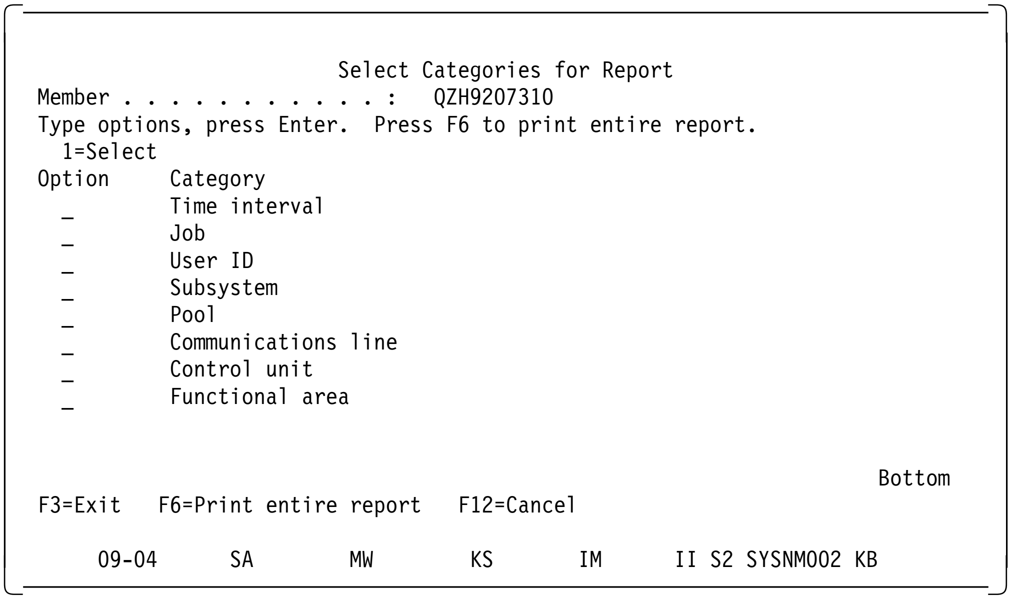
At this point, you may only need the first two commands to determine whether you need to perform problem analysis or not. Type the option on the selected data member on Print Performance Report display.



```
                           Print Performance Report
Library . . . . . .   QPFRZH01
Type option, press Enter.
  1=System report   2=Component report   3=Transaction report   4=Lock report
  5=Job report   6=Pool report   7=Resource report   8=Batch job trace report
Option   Member       Text                                 Date       Time
  _      Q962160900   Data collected 080396 at 0900        08/03/96   09:00:38
  _      Q962130900   Data collected 073196 at 0900        07/31/96   09:00:42
  _      Q962120900   Data collected 073096 at 0900        07/30/96   09:00:43
  _      Q962101447   Data collected 072896 at 0900        07/28/96   14:47:34
  _      Q962090900   Data collected 072796 at 0900        07/27/96   09:00:44


                                                                        Bottom
F3=Exit   F5=Refresh   F11=Work with your spooled output files   F12=Cancel
F15=Sort by member     F16=Sort by text

     10-04      SA        MW        KS       IM      II S2 SYSNM002 KB
```

Figure 21. Print Performance Report Display

You can select a specific category of performance data, or select the entire report.



```
                         Select Categories for Report
Member . . . . . . . . . . . :   QZH9207310
Type options, press Enter.  Press F6 to print entire report.
  1=Select
Option     Category
  _        Time interval
  _        Job
  _        User ID
  _        Subsystem
  _        Pool
  _        Communications line
  _        Control unit
  _        Functional area


                                                                      Bottom
F3=Exit   F6=Print entire report   F12=Cancel

     09-04      SA        MW        KS       IM      II S2 SYSNM002 KB
```

Figure 22. Select Categories Display

The System report and Component report give complete information to evaluate your system performance. To find out whether you need to perform problem analysis or trend analysis, just pay attention to the following items:

- Average response time in the System Report Workload
- Number of transactions for total run time and per hour
- CPU percent for all levels of priority and cumulative. The cumulative value up to and including priority 20 is the value to be compared against the threshold value for CPU for interactive work.
- Number of database/non-database page faults in each storage pool
- Disk, percent full, and utilization of access arms
- Communication lines traffic and IOP utilization

System Report: In the System report, you find the basic set of information to compare against your *performance objectives* and guidelines tables as shown in Table 33 on page 383.

- The System Overview Workload and Resource Utilization part shows what the system workload is and what the cost of doing the workload is. The CPU Utilization shows the percentage of processing unit time used by each job type. According to the guidelines, the interactive CPU utilization should not exceed 81% (for 4-way processors). See Table 33 on page 383 for other CPU types.
- Check the percent of space in use and the utilization on the Disk Utilization part of the report; compare those values to Table 33 on page 383. The Column Ops Per Second and number of disk IOPs installed on the system show whether or not you are overdriving the IOPs. On normal distribution disk operations, each IOP′s average should be between 30 to 60 per second.
- The Avg Util and Max Util column on the Communication part gives you the average and maximum percentage of line capacity used during the measured interval. Compare those values to Table 33 on page 383.

If you find any discrepancy between the System Performance report and the guidelines, go to the Component Report to find out whether you need to do problem analysis on the system performance.

Component Report: The Component report provides information about the same components as the Systems report, but at a greater level of detail.

- Component Interval Activity shows the use of processing unit, disk, and pools at various time intervals. You can check if your system always runs a high transaction rate in all intervals, if the same disk unit appears in the high utilization, or to identify the pool that has the highest fault rate.
- Job Workload Activity gives the activities of each job. You need to perform problem analysis on a particular job if you find that job used most of the disk I/O operation (under column Disk I/O) or CPU utilization (CPU Util).
- In the Pools Storage Activity part, you need to look at column DB faults and Non-DB faults. Compare those values to Table 33 on page 383. Wait-to-ineligible need not be 0 all the time, but must be less than .25 for good performance.
- Disk Activity shows the average disk activity per hour and the capacity of each disk. Batch processing may cause a high utilization of individual disk drives. Batch sequential processing can stay on one drive for some time. If the system is running in dedicated batch, performance is not normally degraded. However, if there are lots of interactive jobs, high disk utilization can indicate a performance problem.
- The Database Journal Summary includes the user journal and system journaling of access paths disk write counts. No guidelines are provided so you must record this information over time to determine any increase in the disk I/O as a result of journaling.

If you still need more data on your current system performance before you decide to analyze, issue the WRKSYSACT command. Refer to Section 9.4, “Reviewing the Measured Data” on page 130 for more information about that command.

Based upon this information, you can decide if there is any problem with the interactive performance of the system.

The batch work of your system should also be taken into consideration when determining whether or not there is a performance problem. If your batch work is not finishing within the expected time, run the Performance Monitor for the batch work time frame.

Chapter 9, “Performance Problem Analysis” on page 127 provides details for analyzing performance data.

If you have no indication of any problems with system performance, proceed to Chapter 8, “Performance Trend Analysis” on page 107.

7.5 Performance Data Conversion

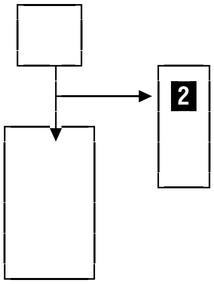
Each new release of the AS/400 system may require conversion of performance data collected on a previous release. Performance data collected with STRPFRMON or WRKPFRCOL from a previous release system needs to be converted before analyzing. The CVTPFRDTA command converts your performance data library from a previous release to a current one. Data conversion may affect the other transaction response times. You may consider submitting it during a low period of CPU utilization.

After all of the performance data has been successfully converted, other Performance Tool functions can be performed on the new release.

Chapter 8. Performance Trend Analysis

This chapter expands on the topic of trend analysis introduced as Stage **2** of the performance management flowchart in Chapter 6, “Performance Management Methodology.”

Performance trend analysis evaluates system performance on a regular basis. Its primary objectives are to understand the performance characteristics or ″health″ of a particular operating environment and identify areas where system capacity may need attention.



8.1.1.1 Major Topics Covered in This Chapter

- Why Do Trend Analysis?
- When Should Trend Analysis Be Done?
- Trend Analysis Methodology
- Performance Graphs
- Trend (Historical) Graphs
- PM/400
- Performance Investigator
- Archiving Performance Data
- BEST/1 Capacity Planning Including Batch Run Time Analysis
- Comparing Extrapolated Data with Historical Data
- Notification of Performance Trends

8.2 Why Do Trend Analysis?

Even if you currently have no performance concerns, it is beneficial to be aware of any potential capacity problems before they occur so that timely corrective action may be taken. One of the key assumptions in BEST/1 is that the workload used as a basis for modelling is not under any severe resource limitations. If you wait until you have resource problems and bottlenecks prior to collecting performance data, the information is not ideal for use with the BEST/1.

Therefore, you should collect performance data regularly and review the data for performance problems. Even if no problems are encountered, you need to determine if there is a pattern or trend in resource utilization and workload that might signal a potential bottleneck in system resource.

8.3 When Should Trend Analysis Be Done?

Trend analysis can be done daily or weekly but probably it is enough to check the trend on a monthly basis. If the information shows a sharp increase in the rate of increase of resource utilization, a more frequent review may be indicated. If a problem is encountered, identifying and resolving the problem takes precedence over trend analysis activity but does not replace it. Trend analysis is an insurance against the sudden discovery of a capacity blow-out.

8.4 Trend Analysis Methodology

Data collected by running the Performance Monitor can occupy a considerable amount of disk space, depending on the duration of the collection period, the time interval for sampling the data, and whether trace data is collected or not.

Once the performance data set is reviewed for potential problems, as discussed in Chapter 4, “AS/400 Performance in a Server Environment,” a subset of the data may be retained for trend analysis. Alternatively, you can periodically save the entire set of data to magnetic media if you feel there may be a need to look at details at a later time. The presumption in this publication is that by constantly measuring and analyzing performance data, there is no need to save this much information.

Utilization trends can be detected in most of the system resources such as CPU, disk, IOP, communications lines, and memory. Increases in workload may also show trends reflected by the number of transactions per hour or by average response times.

8.5 Performance Graphs

This is a function of the AS/400 Performance Tools program product that is used for graphically displaying performance data. It requires performance data to be collected using the STRPFRMON command.

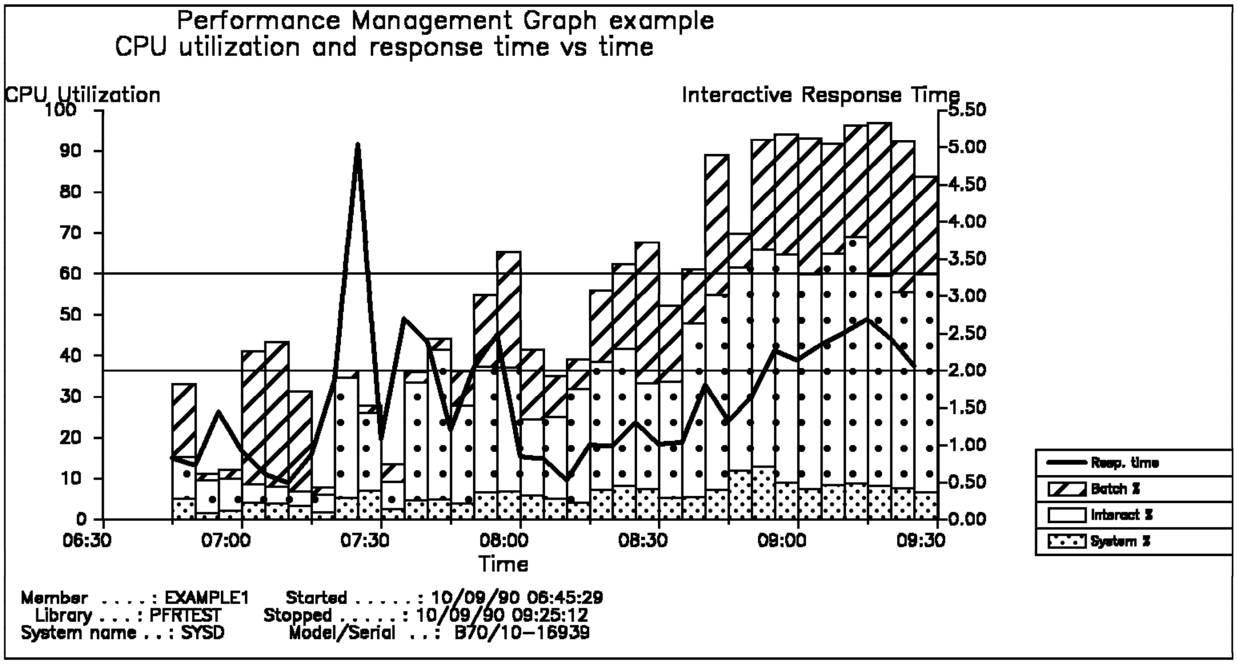
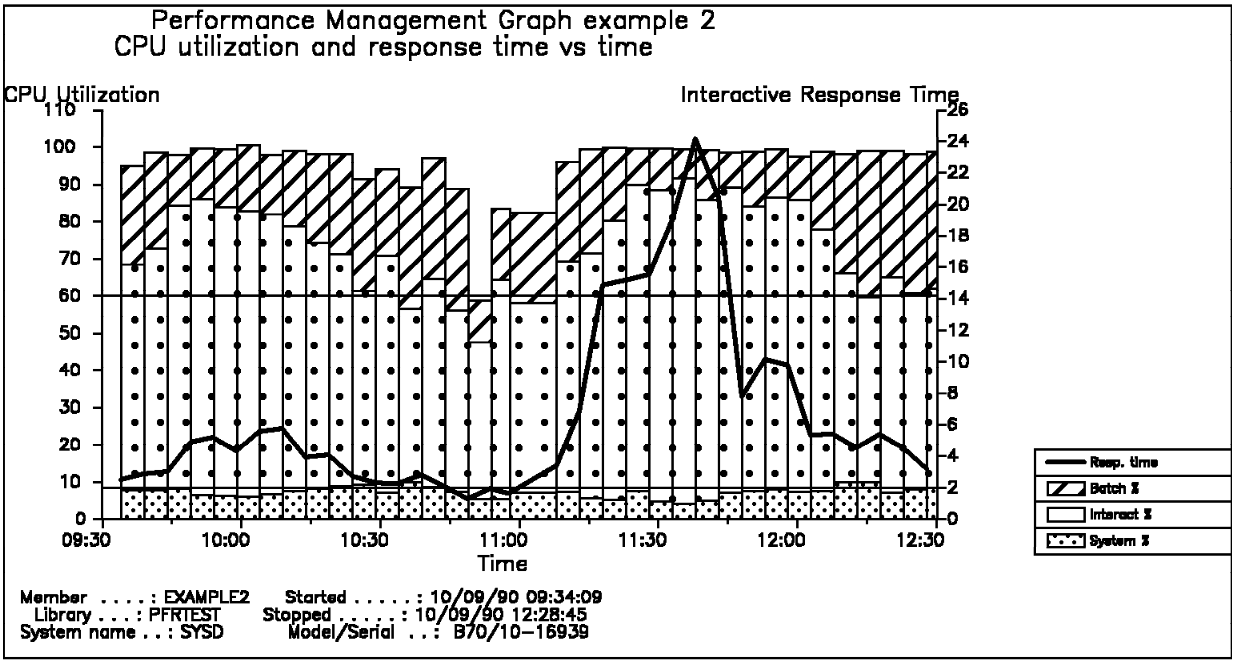


Figure 23. CPU by Type plus Response Time

This graph is an example of how to show performance data in a graphical manner. It shows that the spikes of relatively poor response between 7:20 and 8:00 coincide with the blocked part of the previous query listing, and that the response then settles down and rises gradually to 2.5 seconds at 9:15 as the CPU utilization for interactive work increases to the upper threshold limit of 70%.

Trends are not always as easy to see as this, and in some cases, it may not be possible to see the cause of poor response from the information shown in the profile query or this graph.

In most cases, however, you should be able to detect a trend or relationship between throughput, response time, and one or more fields in the measured profile query. By graphing this file using one record per x-axis graduation, the trend may become more obvious.



```
Member . . . . : EXAMPLE2     Started . . . . . : 10/09/90 09:34:09
  Library . . . : PFRTEST     Stopped . . . . . : 10/09/90 12:28:45
System name . . : SYSD          Model/Serial . . :  B70/10-16939
```

Figure 24. CPU by Type plus Response Time

This graph shows the poor interactive response time between 11:15 and 12:00. The reason for this is not obvious, and more investigation is required, but at least it indicates a band of samples where more detailed analysis should be directed.

It also shows that the CPU percentage utilization for interactive use is consistently high and that the response time improves noticeably when it drops below the 60% mark.

8.6 Trend (Historical) Graphs

This is a function of the AS/400 Performance Tools program product.

The function for displaying historical data with the performance tools requires you to use the CRTHSTDTA command on every measurement member you want included. Menus accessed through the Performance Tools STRPFRT command also provide this function. After this creation of historical data is complete, the collected data may be deleted to save disk space. However, you should consider whether some time period should pass before deleting the data and if it is prudent to save the data to suitable magnetic media.

The DSPHSTGPH command allows you to specify to display (*), to plot (*PLOT), to print on a printer (*PRINT), or to put the graph into an outfile (*OUTFILE) for later use.

The following example shows performance historical data over a one-month period of time. The data includes CPU utilization versus response time, but can actually show many other different performance resources as well. Note that library PFRMGMT (which contains the Automated Performance Management Tools) includes the graphic definitions used in these examples.

The *AS/400 Performance Tools Guide*, SC41-4340, provides more information on the use of performance graphics.

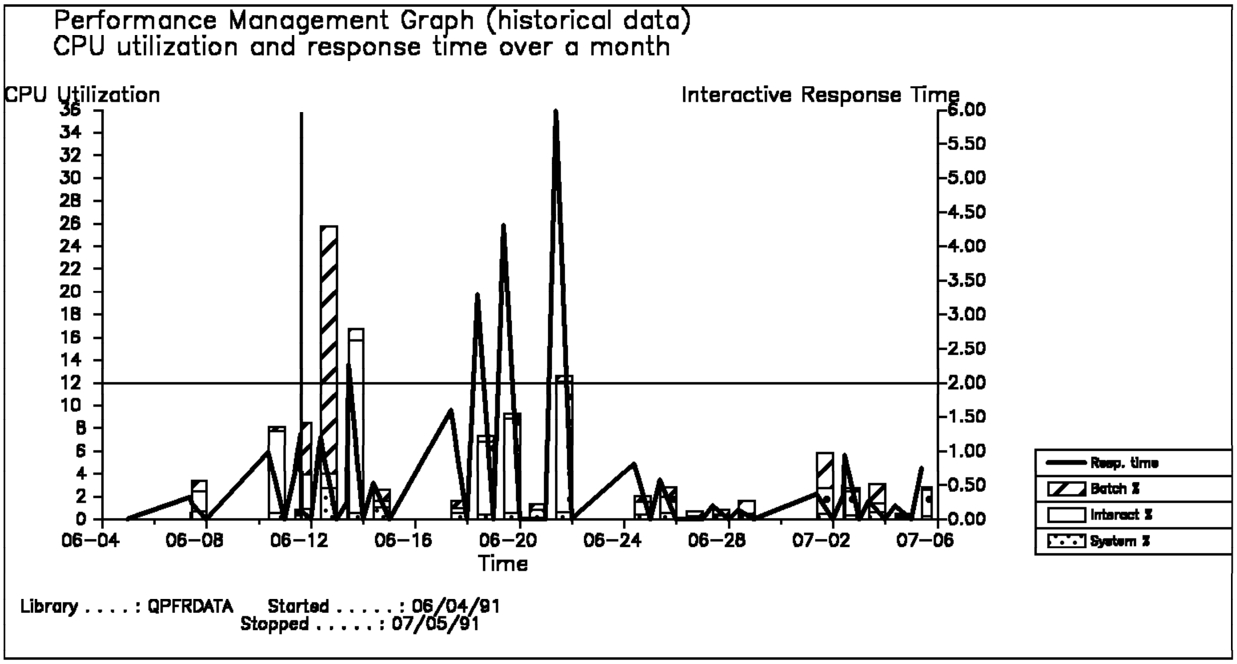


Figure 25. DSPHSTGPH Example Showing CPU Utilization versus Response Time

8.7 Archiving Performance Trend Data

The BEST/1 Capacity Planning tool builds models using performance monitor data, which includes pass-through, Client Access/400, and 5250 emulation under RUMBA and Communications Manager. These models can be used to show a snapshot of performance history.

One way of archiving performance data is to use the scheduler function in PM/400 combined with the IBM SAVLIB command or SAVOBJ command. The PM/400 scheduler collects performance data on a rolling basis. The length of time that data is retained is set by the customer. An example is to set the collection of data for a rolling 10-day period and save the performance library once each week. This means that at any one time, only 10 days of data is stored on the AS/400 system, but there is record of performance data for every day stored in weekly tapes.

8.7.1 Historical Data

Using the CRTHSTDTA command, the information for the historical data is stored in two files and takes a small amount of space compared to actual performance monitor files:

- QAPGHSTD
- QAPGHSTI

The record lengths for these files are small (less than 150 bytes), but for each measurement, there is a few hundred records in both files. The total size (including an access path) might range from 0.5MB to 2MB for a month of data.

PM/400 also summarizes performance data each night, and this is stored in small summary files each night. This process is automated by PM/400 scheduler. It is possible to analyze this data and produce reports, but the PM/400 service offering automates this for you.

8.8 Performance Investigator/400 (PI/400)

Performance Investigator/400 (PI/400) is a utility designed to show short term trends in performance with the ability to set *triggers*, so that when certain conditions occur, an alarm occurs to make the operator aware of this occurrence. It is also possible to program an action in response to the trigger. PI/400 is not designed for medium or longer term performance analysis. It is designed as an operational and system management utility.

More information about this tool can be found in Section 10.3, “Performance Investigator/400 (PI/400)” on page 163.

8.9 BEST/1 Capacity Planning Tool

If you are interested in seeing how the changes to your configuration affect performance, you can do so using the BEST/1 Capacity Planning Tool that was introduced for V2R2 and replaces the MDLSYS Capacity Planner with many new functions.

When analyzing trend performance data, BEST/1 models are created using the actual performance monitor data. BEST/1 models can also be created from a set of provided *predefined workloads* or by explicit user entry of transaction definitions.

Models created from actual performance monitor data offer the most accurate capacity planning results.

8.9.1 BEST/1 Tips

The following list contains various tips or limitations when modeling BEST/1:

- BEST/1 does not model the Set Object Access command or expert cache OS/400 support:

  Performance monitor data collected while these ″tuning″ options are in effect are included in any model created by BEST/1, but there are no options to model with or without a caching option in effect.

- BEST/1 models communication I/O Processor (IOP) and communication line utilization:

However, even though the Integrated PC Server (formerly known as FSIOP) utilization is reported with other communication IOPs in the Performance Tools/400 reports, BEST/1 does not model the Integrated PC Server such as varying cache memory sizes.

There is insufficient experience with LAN Server/400 and Integrated PC Server environments to develop modeling algorithms. Since most of the ″work″ is done in the Integrated PC Server, there is normally little data for BEST/1 to analyze.

- BEST/1 can be used to model the disk hardware write cache support available on the internal 6502, 6512, and 6530 IOPs and the 6501 external IOP:

  The user of BEST/1 must understand the details of the disk caching ″rules″ and manipulate BEST/1 parameter values accordingly. (For details on disk write cache modeling, refer to the *disk* information in Chapter 3, “Factors Affecting Performance” on page 11.)

  It may not be worth the time and effort to change the cache efficiency ratio as BEST/1 has factored into its algorithms the history of lab experience with many sets of customer performance data.

  The write cache is available on newer technology disks such as 6602, 6603, 6605, 6606, 6607, 9337-2xx, and 9337-4xx disks. The I/O characteristics of the application determine the benefit of this cache. If you have collected data from a system with these drive models, you can model the effect of adding cache by changing the cache efficiency number on the controller.

  However, this only has a secondary effect and may not show any changes in transaction response time. The effect of the cache is to reduce I/O which reduces CPU due to I/O and may not directly show in up modeled response numbers. If the change is drastic, you may see changes in the disk arm report, but if it is a large system with lots of arms, small changes are hard to find.

  Remember, as you modify the cache efficiency number, you are making theoretical changes in the data being processed by the application, such as more random processing of records written to disk. In the real environment, you may have no control of what records are processed and their physical location on the disk arms.

- The *BATCHJOB workload attribute eases batch run-time modeling:

  In V2R3, the *BATCHJOB workload attribute was introduced to change some of the internal algorithms used by BEST/1 for batch modeling.
  Non-interactive job run-time modeling still requires the BEST/1 user to convert differences in the ″before″ and ″after″ transaction counts into units of time.

  Batch transaction counts and response times should be treated only as an indication of throughput. Varying CPU or other hardware resources either increases or decreases transaction rates that are applied to manual calculations of run time.

  For example, using a model created from actual performance monitor data showed 8 000 non-interactive transactions. Assume that the user recorded this number of transactions completed in four hours.

  Change CPU or disk configuration and do model analysis. Presume the number of non-interactive transactions for that workload is now 10 000. This is a 20% improvement. Therefore, the four-hour job should be reduced by approximately 20% so the new predicted run time is 3.2 hours assuming the run-time environment, including the number of records processed by the application, remained the same.

Note that the model must first be created with default workload attributes. After the user has verified the accuracy of the model, the appropriate workload can be changed to have the *BATCHJOB attribute. Then the ″what-if″ configuration change modeling can be performed.

Considerations for *BATCHJOB modeling:

- The batch jobs modeled should be few and long running.
- The batch jobs run throughout the entire interval.
- The batch jobs are doing as much work as the system allows.
- Amount of work done by batch decreases if other higher priority activity is increased, and vice versa.
- Batch jobs cannot be running in a pool with interactive work.

- V3R6 supports 24 new predefined workloads based upon PM/400 performance data:

  These new workloads include 22 industry workloads and two client/server workloads that are described in Appendix B of the *BEST/1 Capacity Planning Tool Guide*, SC41-4341.

  Predefined workloads should only be used when there are no measurements of future workloads or new applications under development. Be aware that when you use these predefined workloads, the margin for error is greatly increased.

- New page fault guidelines and Disk IOP guidelines:

  V3R6 page fault rates in Appendix A, “Guidelines for Interpreting Performance Data” on page 383 are supported.

8.10 Examples for BEST/1, Migrating to PowerPC AS/400 Systems

In October 1995, BEST/1 received PTFs to model all of the announced PowerPC AS/400 system models and features. This exercise is based on V3R1 and V3R6 BEST/1 for PowerPC modeling. PTF SF29757 must be applied in your V3R1 AS/400 system. This PTF may be superseded during 1996, but can be used as a reference when discussing the support with IBM Service. To be current with all of the available PTFs for V3R6 BEST/1, please apply the latest available cumulative package for this version.

You can tell if this PTF is on your system by doing DSPPTF for 5763-PT1 for V3R1 or 5716-PT1 for the V3R6 Performance Tools/400 licensed program. Alternatively, you can start BEST/1 and view the CPU models through the BEST/1 Configuration menu option ″Change CPU and other resource values″. With the PTF installed, you see RPR values for the 530 and 53S system features.

While modeling a system, you should understand unique BEST/1 parameters or parameter values (″conversion factors″) used to model PowerPC systems from Performance Monitor data collected on a non-PowerPC system. BEST/1 uses these conversion factors while upgrading to a PowerPC system. For V3R1, these BEST/1 factors are ″hard-coded″ and you cannot change them. Table 9 shows these factors.

Table 9. PowerPC Conversion Factors

Workload Type	CPU Time	Working Set Size	I/O Counts
*NORMAL	1.00	2.00	1.00
*TRNNORM1 1	.50	1.50	1.00
*TRNNORM2 2	1.00	2.30	1.00
*TRNNORM3 3	1.50	5.00	1.00
*BATCHJOB	1.00	2.00	1.00
*TRNBAT1 1	.50	1.50	1.00
*TRNBAT2 2	1.00	2.30	1.00
*TRNBAT3 3	1.50	5.00	1.00

For all models, BEST/1 defaults to workload types of either *NORMAL or *BATCHJOB and the conversion factors are shown for these workloads in Table 9. These are CPU time of 1.00, Working Set Size of 2.00, and disk I/O counts of 1.00.

For the PowerPC modeling exercises in this chapter, we use the *NORMAL conversion factors. In almost 100% of all modeling of client/server applications or production mode interactive 5250-based applications, *NORMAL provides the correct ″what-if″ results. It is only in cases such as an excessively high CPU application, for example, an ILE C program doing financial modeling or in Original Program Model (OPM) compilation workloads that the BEST/1 user must manually change the workload type from *NORMAL.

You can manually upgrade your system to a PowerPC processor model or let BEST/1 suggest that you upgrade when necessary. The following exercises show you how to use both methods.

The new *TRNNORMn and TRNBATn BEST/1 workload types specifically address workloads that are highly CPU intensive 1, highly disk I/O intensive 2, or dedicated to creation (compilation) of Original Program Model (OPM) 3 programs.

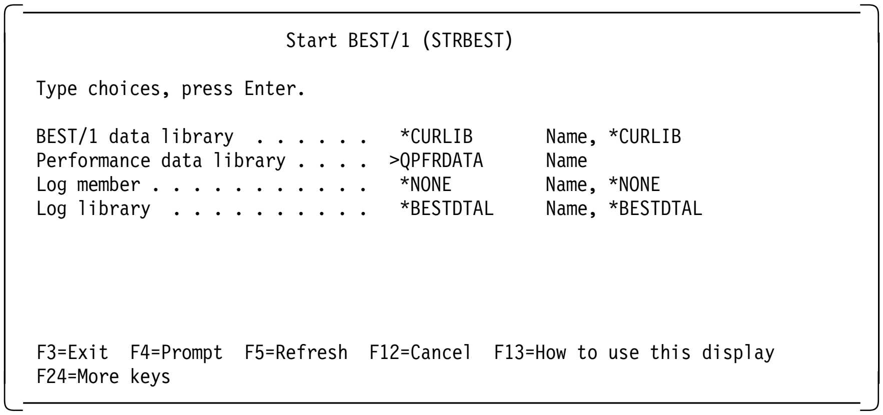
8.10.1 Exercise 1: Manually Upgrading to a PowerPC System

We use the term ″manual upgrade″ under BEST/1 when the BEST/1 user manually selects the CPU model and associated main storage and disk hardware configuration. BEST/1 also supports an ″automatic upgrade″ to new hardware if that option is chosen.

We begin with a manual upgrade exercise so we can better see how far the manually selected hardware resources can take the workload growth beyond the guideline values used by BEST/1.

In this sample, we use data that was collected by Performance Monitor in a 9406-D60 Model:

1. Sign on to the system.
2. Start BEST/1 using the command STRBEST and F4 to prompt for input.



```
                         Start BEST/1 (STRBEST)

 Type choices, press Enter.

 BEST/1 data library  . . . . . .   *CURLIB       Name, *CURLIB
 Performance data library . . . .  >QPFRDATA      Name
 Log member . . . . . . . . . . .   *NONE         Name, *NONE
 Log library  . . . . . . . . . .   *BESTDTAL     Name, *BESTDTAL





 F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
 F24=More keys
```

3. Select menu option 1:

   `1. Work with BEST/1 models`

   In this step, take option 1 again to create and calibrate the model you need. (Information about how to create and calibrate models can be found in *BEST/1 Capacity Planning Tool*, SC41-3341.)

4. Once your model is created and properly calibrated, select option 5=Work with to work with the model previously saved:

   ```
   Opt  Model
   5    Your_Model
   ```

5. Select menu option 10:

   `10. Configuration menu`

6. Select menu option 1:

   `1. Change CPU and other resource values`

7. Change CPU model to the adequate PowerPC model (that is, 500-2141 with RPR = 8.30).

8. Change the main storage size to 96MB because 80MB (which was the amount of memory the D60 had) is not valid for the 2150 CPU Model.

9. Press Enter.

10. Press F17=Re-scale pool sizes.

    Note

    BEST/1 assumes a minimum machine pool size of 16MB when you select a PowerPC model.

11. Go back to the Configuration menu and select menu option 2:

    `2. Work with disk resources`

12. Delete all of the disk resources that are not compatible with the 2141 model.

    Note

    If you do not know the resources that are not compatible, you can wait until BEST/1 sends you the warning messages while trying to work with the model. You may alternatively press F13=Check configuration on the Configuration menu display and BEST/1 validates the hardware configuration immediately.

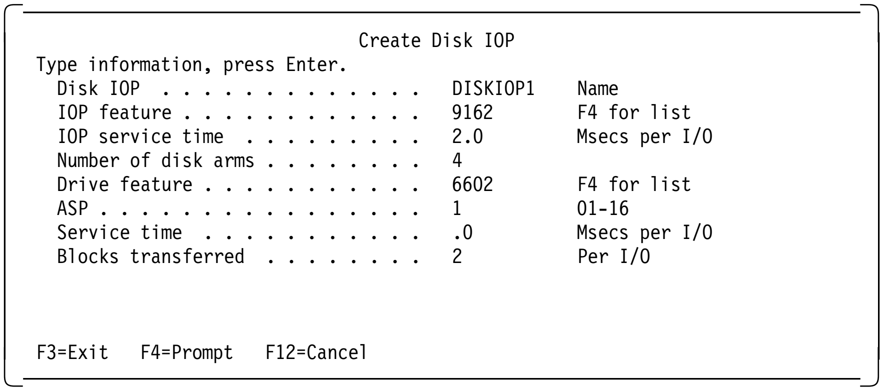
    Select option 4=Delete for the following resources:

- 2615 Disk IOP
- 6112 Disk IOP

13. Press Enter to confirm delete of disk resources:

    This also deletes disk controllers and arms.

14. Press F6=Create disk IOP if it is necessary.



```
                            Create Disk IOP
Type information, press Enter.
  Disk IOP  . . . . . . . . . . . . .  DISKIOP1    Name
  IOP feature . . . . . . . . . . . .  9162        F4 for list
  IOP service time  . . . . . . . . .  2.0         Msecs per I/O
  Number of disk arms . . . . . . . .  4
  Drive feature . . . . . . . . . . .  6602        F4 for list
  ASP . . . . . . . . . . . . . . . .  1           01-16
  Service time  . . . . . . . . . . .  .0          Msecs per I/O
  Blocks transferred  . . . . . . . .  2           Per I/O



F3=Exit   F4=Prompt   F12=Cancel
```

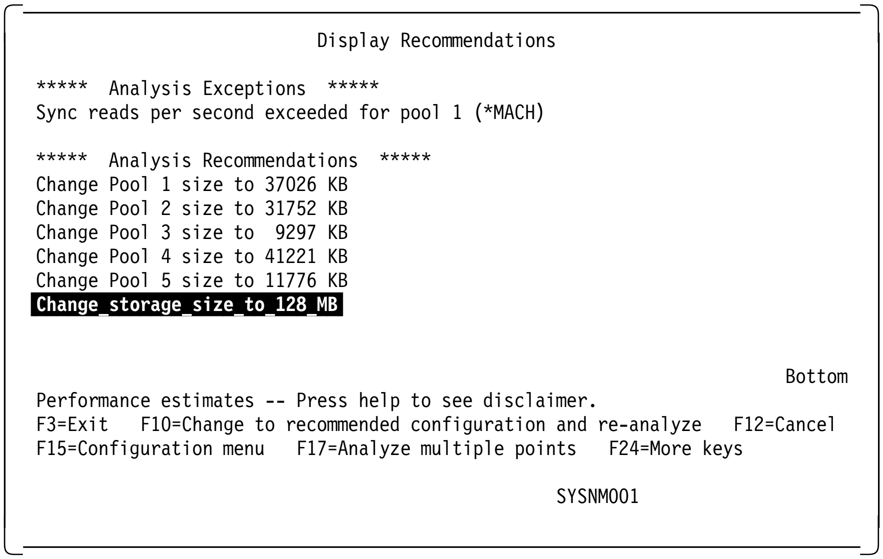
    In this example, the Create Disk IOP display shows the first of the IOPs to create. Use the same command to create all the ″disk-capable″ IOPs you need to complete your new DASD configuration.

15. Press Enter twice to go back to the Work with BEST/1 Model menu.

16. Select menu option 6:

    ```
    6. Analyze current model and give recommendations
    ```

17. Select option 5=Display in the Analysis Summary report and the Recommendations Report. Note in the Recommendations Report, BEST/1 recommends an upgrade to 128MB.



```
                        Display Recommendations

*****  Analysis Exceptions  *****
Sync reads per second exceeded for pool 1 (*MACH)

*****  Analysis Recommendations  *****
Change Pool 1 size to 37026 KB
Change Pool 2 size to 31752 KB
Change Pool 3 size to  9297 KB
Change Pool 4 size to 41221 KB
Change Pool 5 size to 11776 KB
Change_storage_size_to_128_MB


                                                                  Bottom
Performance estimates -- Press help to see disclaimer.
F3=Exit   F10=Change to recommended configuration and re-analyze   F12=Cancel
F15=Configuration menu   F17=Analyze multiple points   F24=More keys

                                             SYSNM001
```

BEST/1 uses the CISC to RISC conversion factors for workload type *NORMAL shown in Table 9 on page 115. In some *NORMAL cases, BEST/1 recommends twice the current main storage and an additional 16MB for a PowerPC model and a minimum of 16MB is required in the system machine pool. The only situation where this is not followed is when there is an excess of main storage prior to upgrading to the PowerPC model. In the current example, because a Model D60 was using a small amount of main storage, BEST/1 might estimate that the allocation of 16MB of machine pool storage and sufficient storage for a current user′s workload is easily contained in going from 96MB (originally 80MB) to 128MB of main storage. If our measured performance data had shown page faulting at the high end of ″acceptable″ according to the *Work Management Guide* range, BEST/1 would have recommended larger main storage.

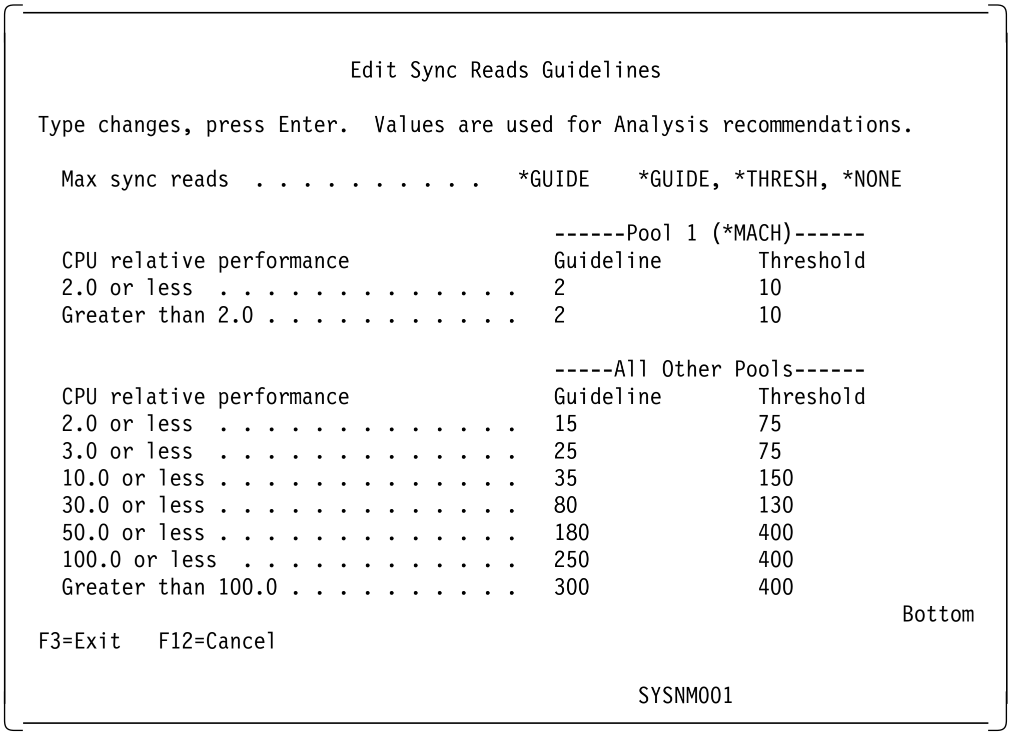
Note on Machine Pool (Pool 1 (*MACH)) Recommendation

BEST/1 uses ″Synchronous disk reads per second″ as an approximation of page faults per second. BEST/1′s Edit Synchronous Reads menu (from the ″More Best/1 options″ option) associates the page faults per second values with guideline and threshold synchronous reads values shown in Figure 26.

Our experience indicates that because synchronous reads include more than just page faults, BEST/1 has a tendency to recommend increasing pool sizes somewhat prematurely. Also, for the machine pool, we believe the guideline values should be two for **all** system models (as shown for ″2.0 or less″ under CPU relative performance in the figure).

Since our review of the System Report and Component Report indicated page faulting in our performance data was close to the *GUIDE values, we are not concerned about needing more storage per the recommendations previously shown.

The topic of Synchronous Read adjustment is discussed in further detail in the *BEST/1 Capacity Planning Tool* manual, SC41-3341.



```
                          Edit Sync Reads Guidelines

 Type changes, press Enter.  Values are used for Analysis recommendations.

   Max sync reads  . . . . . . . . . .   *GUIDE    *GUIDE, *THRESH, *NONE

                                            ------Pool 1 (*MACH)------
   CPU relative performance                 Guideline       Threshold
   2.0 or less  . . . . . . . . . . . . .   2               10
   Greater than 2.0 . . . . . . . . . . .   2               10

                                            -----All Other Pools------
   CPU relative performance                 Guideline       Threshold
   2.0 or less  . . . . . . . . . . . . .   15              75
   3.0 or less  . . . . . . . . . . . . .   25              75
   10.0 or less . . . . . . . . . . . . .   35              150
   30.0 or less . . . . . . . . . . . . .   80              130
   50.0 or less . . . . . . . . . . . . .   180             400
   100.0 or less  . . . . . . . . . . . .   250             400
   Greater than 100.0 . . . . . . . . . .   300             400
                                                                        Bottom
 F3=Exit   F12=Cancel

                                                SYSNM001
```

Figure 26. BEST/1 Edit Synchronous Read Guidelines

18. Press Enter to go back to the Work with BEST/1 Model menu and select menu option 10:

    `10. Configuration menu`

19. Select menu option 1:

    `1. Change CPU and other resource values`

20. Change the Main Storage size to 128MB as recommended.

21. Press Enter.

22. Press F17=Re-scale pool sizes.

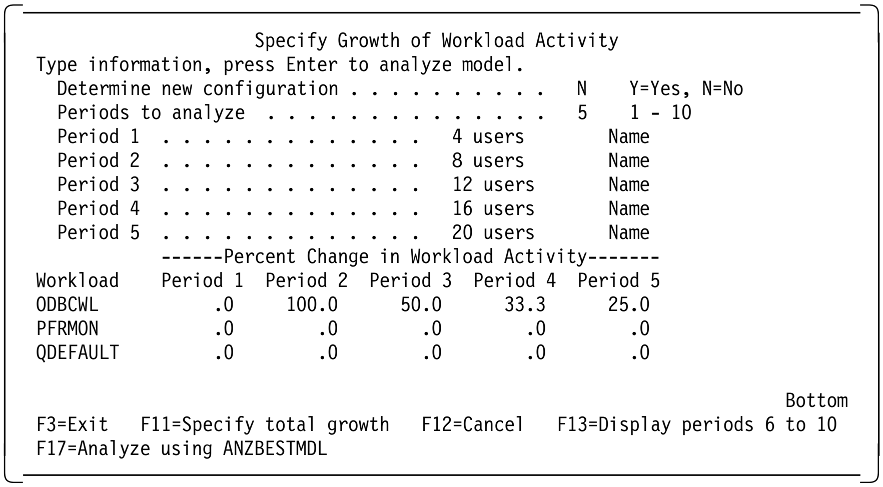
23. Press Enter twice to go back to the Work with BEST/1 Model and select menu option 7:

    `7. Specify workload growth and analyze model.`

24. In this case, we consider only a client/server workload growth.

25. Press F11=Specify growth by workload.

26. Enter the parameters as shown in the following display:



```
                     Specify Growth of Workload Activity
Type information, press Enter to analyze model.
  Determine new configuration . . . . . . . . . .   N    Y=Yes, N=No
  Periods to analyze  . . . . . . . . . . . . . .   5    1 - 10
  Period 1  . . . . . . . . . . . . .  4 users       Name
  Period 2  . . . . . . . . . . . . .  8 users       Name
  Period 3  . . . . . . . . . . . . .  12 users      Name
  Period 4  . . . . . . . . . . . . .  16 users      Name
  Period 5  . . . . . . . . . . . . .  20 users      Name
            ------Percent Change in Workload Activity-------
Workload    Period 1  Period 2  Period 3  Period 4  Period 5
ODBCWL            .0     100.0      50.0      33.3      25.0
PFRMON            .0        .0        .0        .0        .0
QDEFAULT          .0        .0        .0        .0        .0

                                                                 Bottom
F3=Exit   F11=Specify total growth   F12=Cancel   F13=Display periods 6 to 10
F17=Analyze using ANZBESTMDL
```

27. Press Enter to analyze.

28. On Work with Results menu, review every result by typing 5=Display option for ″All of the above″.

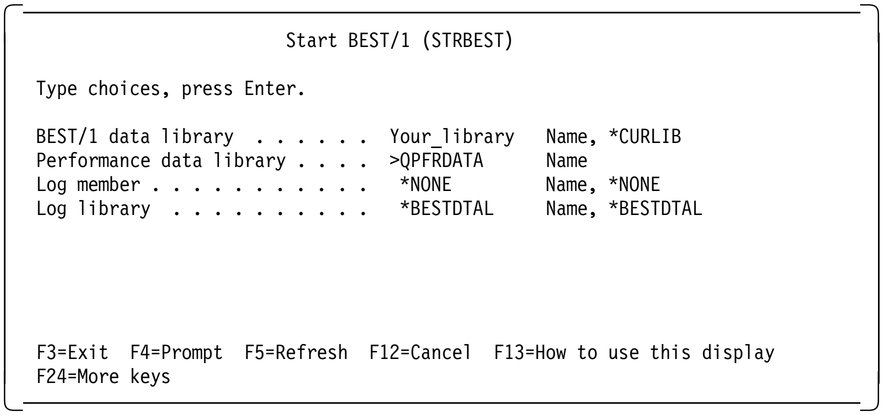
29. On the Work with Results menu, review every result by typing 5=Display option for ″All of the above″.

Especially review the Analysis Summary and Workload Reports.

8.10.2 Automatic Upgrade to a PowerPC System

Under the BEST/1 ″automatic upgrade″ process, we can let BEST/1 use its guideline values to automatically add hardware, perhaps multiple times, if workload growth is significant over the specified number of periods. Automatic upgrade is often used if significant workload growth rates are expected. This is because the manual upgrade approach is not as flexible when resources may need to be upgraded several times.

1. Start BEST/1 using the command STRBEST and F4 to prompt for input.



```
                          Start BEST/1 (STRBEST)

Type choices, press Enter.

BEST/1 data library  . . . . . .  Your_library   Name, *CURLIB
Performance data library . . . .  >QPFRDATA      Name
Log member . . . . . . . . . . .   *NONE         Name, *NONE
Log library  . . . . . . . . . .   *BESTDTAL     Name, *BESTDTAL






F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Press Enter.

2. Select menu option 60:

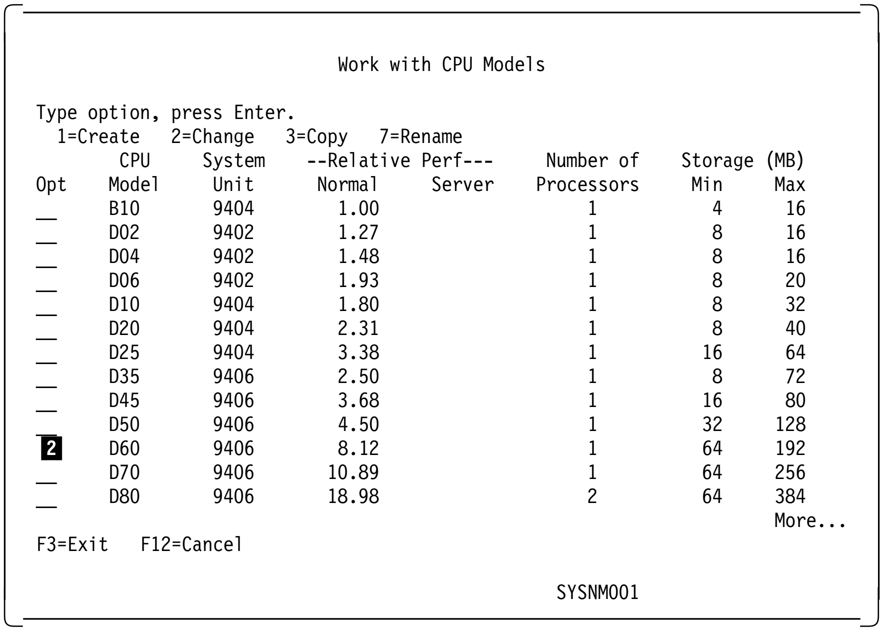
   `60. More BEST/1 options`

3. Select menu option 10:

   `10. Hardware characteristics menu`

4. Select menu option 1:

   `1. Work with CPU models`

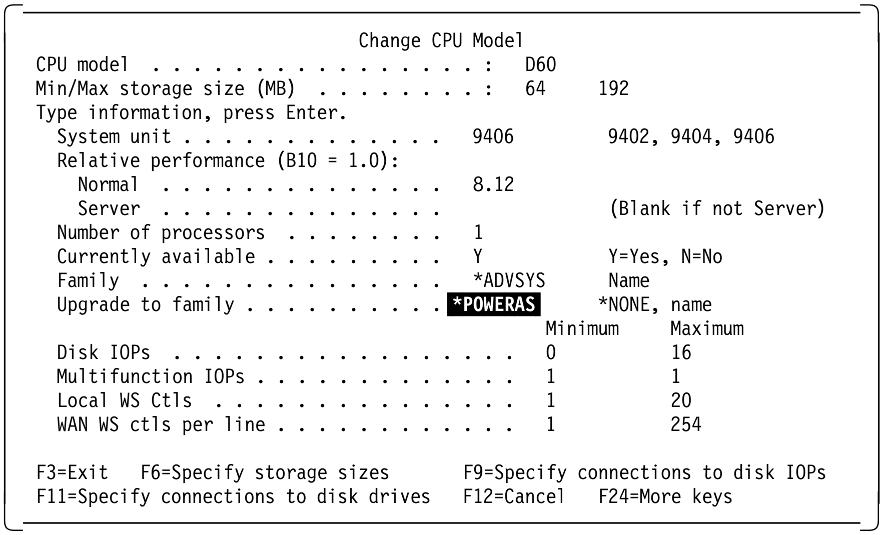


```
                               Work with CPU Models

Type option, press Enter.
  1=Create   2=Change   3=Copy   7=Rename
       CPU     System    --Relative Perf---     Number of     Storage (MB)
Opt   Model     Unit       Normal     Server    Processors      Min     Max
__     B10      9404        1.00                    1             4      16
__     D02      9402        1.27                    1             8      16
__     D04      9402        1.48                    1             8      16
__     D06      9402        1.93                    1             8      20
__     D10      9404        1.80                    1             8      32
__     D20      9404        2.31                    1             8      40
__     D25      9404        3.38                    1            16      64
__     D35      9406        2.50                    1             8      72
__     D45      9406        3.68                    1            16      80
__     D50      9406        4.50                    1            32     128
 2     D60      9406        8.12                    1            64     192
__     D70      9406       10.89                    1            64     256
__     D80      9406       18.98                    2            64     384
                                                                       More...
F3=Exit   F12=Cancel

                                                    SYSNM001
```

5. Select option 2=Change in the CPU model that you have in your base model, in our case a D60. You have to change ″Upgrade to family″ to ***POWERAS** or ***POWERSRV** for the CISC CPU model in your current BEST/1 model to permit BEST/1 to automatically upgrade to a PowerPC system. If you do not change this, BEST/1 upgrades to a ″currently available″ non-PowerPC system when doing an automatic upgrade.

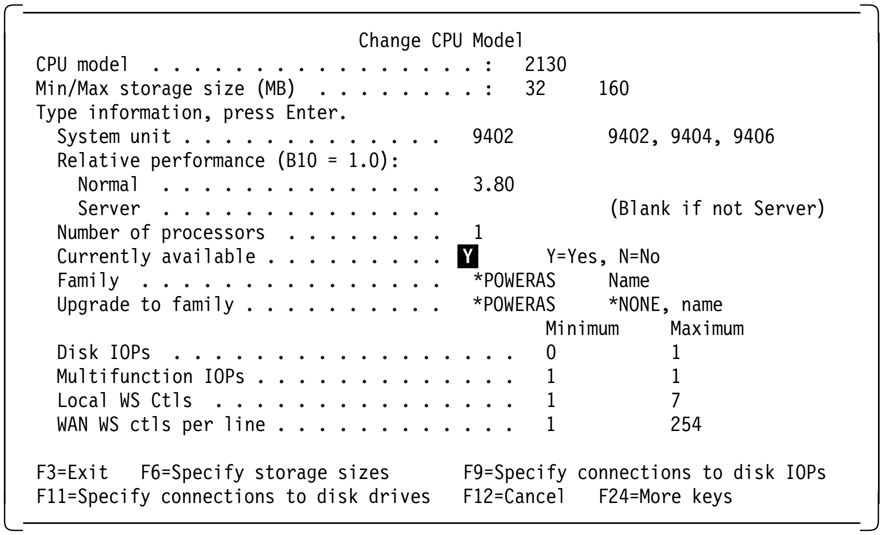
   You should specify ***POWERAS** if you want BEST/1 to consider the traditional models within the PowerPC system family, or ***POWERSRV** if you want BEST/1 to consider server models within the PowerPC system family.

   In this exercise, we use *POWERAS as the D60 is a traditional model.



```
                               Change CPU Model
CPU model  . . . . . . . . . . . . . . . . . :   D60
Min/Max storage size (MB)  . . . . . . . . . :   64     192
Type information, press Enter.
  System unit . . . . . . . . . . . . .   9406         9402, 9404, 9406
  Relative performance (B10 = 1.0):
    Normal  . . . . . . . . . . . . . .   8.12
    Server  . . . . . . . . . . . . . .                (Blank if not Server)
  Number of processors  . . . . . . . .   1
  Currently available . . . . . . . . .   Y            Y=Yes, N=No
  Family  . . . . . . . . . . . . . . .   *ADVSYS      Name
  Upgrade to family . . . . . . . . . .  *POWERAS     *NONE, name
                                                 Minimum     Maximum
  Disk IOPs  . . . . . . . . . . . . . . . . .   0           16
  Multifunction IOPs . . . . . . . . . . . . .   1           1
  Local WS Ctls  . . . . . . . . . . . . . . .   1           20
  WAN WS ctls per line . . . . . . . . . . . .   1           254

F3=Exit   F6=Specify storage sizes       F9=Specify connections to disk IOPs
F11=Specify connections to disk drives   F12=Cancel   F24=More keys
```

6. Page forward on the CPU models list and make sure that all of the PowerPC Advanced Series models have the ″Currently available″ parameter set to Y (Yes).



```
                               Change CPU Model
CPU model  . . . . . . . . . . . . . . . . . :   2130
Min/Max storage size (MB)  . . . . . . . . . :   32     160
Type information, press Enter.
  System unit . . . . . . . . . . . . .   9402         9402, 9404, 9406
  Relative performance (B10 = 1.0):
    Normal  . . . . . . . . . . . . . .   3.80
    Server  . . . . . . . . . . . . . .                (Blank if not Server)
  Number of processors  . . . . . . . .   1
  Currently available . . . . . . . . .  Y        Y=Yes, N=No
  Family  . . . . . . . . . . . . . . .   *POWERAS     Name
  Upgrade to family . . . . . . . . . .   *POWERAS     *NONE, name
                                                 Minimum     Maximum
  Disk IOPs  . . . . . . . . . . . . . . . . .   0           1
  Multifunction IOPs . . . . . . . . . . . . .   1           1
  Local WS Ctls  . . . . . . . . . . . . . . .   1           7
  WAN WS ctls per line . . . . . . . . . . . .   1           254

F3=Exit   F6=Specify storage sizes       F9=Specify connections to disk IOPs
F11=Specify connections to disk drives   F12=Cancel   F24=More keys
```

7. Press Enter to go back to the BEST/1 for the AS/400 menu.

8. Select menu option 1:

   ```
   1. Work with BEST/1 models
   ```

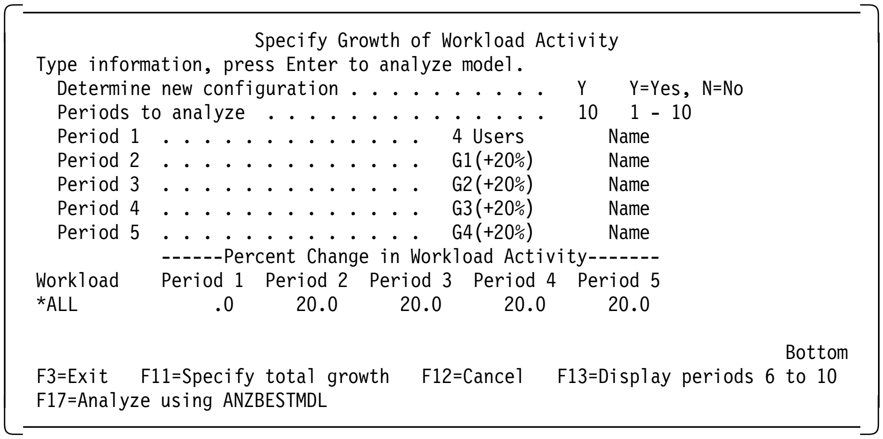
9. Select option 5=Work with, to work with the (calibrated) model previously saved:

   ```
   Opt  Model
   5    YOUR_MODEL
   ```

10. Select menu option 7:

    ```
    7. Specify workload growth and analyze model.
    ```

11. Specify the workload growth as shown in the following display:



```
                     Specify Growth of Workload Activity
Type information, press Enter to analyze model.
  Determine new configuration . . . . . . . . . .   Y    Y=Yes, N=No
  Periods to analyze  . . . . . . . . . . . . . .   10   1 - 10
  Period 1  . . . . . . . . . . . . .   4 Users       Name
  Period 2  . . . . . . . . . . . . .   G1(+20%)      Name
  Period 3  . . . . . . . . . . . . .   G2(+20%)      Name
  Period 4  . . . . . . . . . . . . .   G3(+20%)      Name
  Period 5  . . . . . . . . . . . . .   G4(+20%)      Name
            ------Percent Change in Workload Activity-------
Workload    Period 1  Period 2  Period 3  Period 4  Period 5
*ALL             .0      20.0      20.0      20.0      20.0


                                                                  Bottom
F3=Exit   F11=Specify total growth   F12=Cancel   F13=Display periods 6 to 10
F17=Analyze using ANZBESTMDL
```

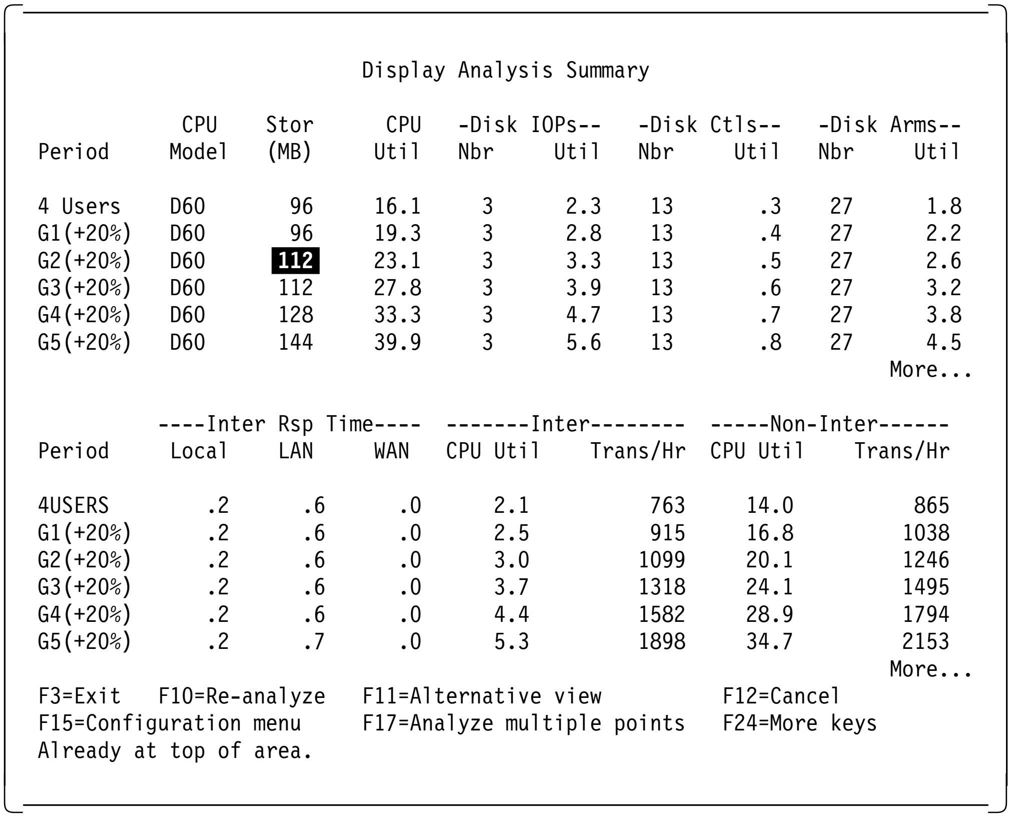
Note

Now we are increasing all of the workloads instead of only our user-defined workload ODBCWL. This is done just to create enough resource utilization increase to cause BEST/1 to automatically upgrade the hardware.

12. Select option 5=Display:

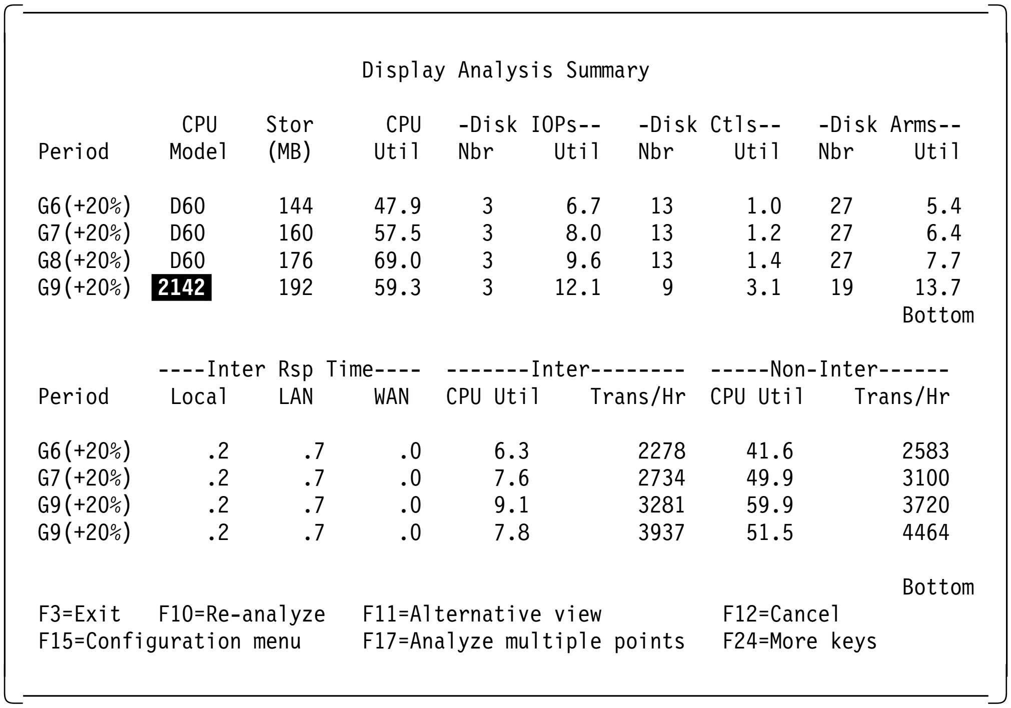
```
Opt      Report Name
5        Analysis Summary
```

Figure 27 and Figure 28 show both pages of the upper portion of the Display Analysis Summary report. Note the added main storage for the D60 during the growth periods and the upgrade to the PowerPC model during period 10 (the model 2142 with RPR of 11.4 compared to the D60 RPR of 8.12). D60 main storage also increased from the original 80MB (96MB) to 192MB.



```
                           Display Analysis Summary

           CPU     Stor      CPU   -Disk IOPs--   -Disk Ctls--   -Disk Arms--
Period     Model   (MB)      Util  Nbr     Util   Nbr     Util   Nbr     Util

4 Users    D60       96      16.1   3       2.3   13        .3   27       1.8
G1(+20%)   D60       96      19.3   3       2.8   13        .4   27       2.2
G2(+20%)   D60      112      23.1   3       3.3   13        .5   27       2.6
G3(+20%)   D60      112      27.8   3       3.9   13        .6   27       3.2
G4(+20%)   D60      128      33.3   3       4.7   13        .7   27       3.8
G5(+20%)   D60      144      39.9   3       5.6   13        .8   27       4.5
                                                                      More...

          ----Inter Rsp Time----  -------Inter--------  -----Non-Inter------
Period     Local    LAN      WAN  CPU Util    Trans/Hr  CPU Util    Trans/Hr

4USERS        .2       .6      .0      2.1         763      14.0         865
G1(+20%)      .2       .6      .0      2.5         915      16.8        1038
G2(+20%)      .2       .6      .0      3.0        1099      20.1        1246
G3(+20%)      .2       .6      .0      3.7        1318      24.1        1495
G4(+20%)      .2       .6      .0      4.4        1582      28.9        1794
G5(+20%)      .2       .7      .0      5.3        1898      34.7        2153
                                                                      More...
F3=Exit   F10=Re-analyze   F11=Alternative view          F12=Cancel
F15=Configuration menu     F17=Analyze multiple points   F24=More keys
Already at top of area.
```

Figure 27. Calibrated Model, Automatic Growth Display 1



```
                           Display Analysis Summary

           CPU     Stor      CPU   -Disk IOPs--   -Disk Ctls--   -Disk Arms--
Period     Model   (MB)      Util  Nbr     Util   Nbr     Util   Nbr     Util

G6(+20%)   D60      144      47.9   3       6.7   13       1.0   27       5.4
G7(+20%)   D60      160      57.5   3       8.0   13       1.2   27       6.4
G8(+20%)   D60      176      69.0   3       9.6   13       1.4   27       7.7
G9(+20%)  2142      192      59.3   3      12.1    9       3.1   19      13.7
                                                                       Bottom

          ----Inter Rsp Time----  -------Inter--------  -----Non-Inter------
Period     Local    LAN      WAN  CPU Util    Trans/Hr  CPU Util    Trans/Hr

G6(+20%)      .2       .7      .0      6.3        2278      41.6        2583
G7(+20%)      .2       .7      .0      7.6        2734      49.9        3100
G9(+20%)      .2       .7      .0      9.1        3281      59.9        3720
G9(+20%)      .2       .7      .0      7.8        3937      51.5        4464

                                                                       Bottom
F3=Exit   F10=Re-analyze   F11=Alternative view          F12=Cancel
F15=Configuration menu     F17=Analyze multiple points   F24=More keys
```

Figure 28. Calibrated Model, Automatic Growth Display 2

8.11 Manual Batch Run-Time Estimation

Batch run-time modeling is possible under BEST/1 as discussed in Section 8.9.1, “BEST/1 Tips” on page 112. Other batch run-time modeling tools are discussed in Appendix C, “IBM Internal Use Only Tools/Documents” on page 393. This section presents a formula for manually predicting run time based on Performance Tools/400 reports, relative CPU RIP ratings, and disk service times.

For relative CPU processor speed RIP ratings, use the tables in Section A.1, “System Capacities for AS/400 PowerPC Advanced Systems” on page 384.

For disk service times, use the average disk access times shown in the Performance Tools/400 System Report.

The basic formula is:

```
Job Run Time = CPU Time + Disk I/O Time + Other Wait Time
```

This equation does not take into account the effect of database or other object contention. Neither does it include the impact of other jobs running concurrently on the system and their job priorities relative to the job being modeled.

Updating the basic equation to include CPU queuing, disk queuing, and job workload characteristics results in the expanded batch run-time equation:

```
Job Run Time =
    Number of LDIO x ((CPUQM x CPU/LDIO) + (PDIO/LDIO x DISKRESP))
```

To estimate the effect of another processor on job run time, multiply the CPU/LDIO parameter by the quotient of the base model′s relative CPU value divided by the new model′s relative CPU value. For example, to estimate a Model B70 time on a Model D70, multiply the B70 CPU/LDIO value by 0.62 as follows:

```
              6.8  (B70 relative CPU)
CPU/LDIO x   -----                      = 0.62
             10.9  (D70 relative CPU)

LDIO:      Logical Disk I/O from Resource Utilization Expansion
           section of the System Report.
CPUQM:     CPU Queuing Multiplier (see the following formula).
CPU:       Total CPU time used by the job from the Workload
           section of the System Report.
PDIO:      Physical Disk I/O from Resource Utilization Expansion
           section of the System Report or the WRKDSKSTS command.
DISKRESP:  Disk Response (see the following formula).

Formulas:

   CPU Queuing Calculation:
                                1
   CPU QM:   --------------------------------------------
                      ----                      ---- **P
                      |                  N-1       |
                  1 - |   ( U1 + (U2 x ----- ) )   |
                      |                   N        |
                      ----                      ----

   where U1 = CPU utilization for all jobs at a higher priority.
         U2 = CPU utilization for all jobs at the current priority
              level.
         N  = number of jobs competing concurrently for the CPU at
```

```
                the current priority level.
           P  = number of processors (E95 = 4, E90 = 3, E80 and D80
                =2, all others = 1)


     Disk Utilization =
              (Physical Disk OPS per second x Base Disk Service Time)
        100 x -------------------------------------------------------
                               Number of  Disk Arms

     Physical Disk OPS per Second:  Measured with Performance Tools
                 using Disk Utilization section of the System Report.

     DISKRESP = Base Disk Service Time x DISKQM
```

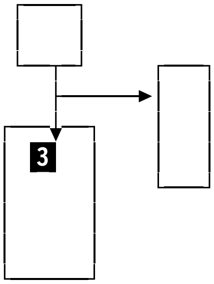
8.12 Notification of Performance Trends

It is important to advise management of the results of a trend analysis. Upward trends in workload and resource utilization must be highlighted so that management may take remedial action in time, either to control the growth in requirements or take steps to acquire additional hardware to support the increased demand for system resources. Predictive reports and graphs can be used to demonstrate the impact of:

1. Making no changes to the system.
2. Installing recommended upgrades using the facilities of BEST/1 Capacity Planning Tool.

Chapter 9. Performance Problem Analysis

This chapter addresses stage 3 of the performance management flowchart block of activity shown in ***Performance Management Methodology*** Performance Problem Analysis.



This chapter provides assistance on how to analyze a performance problem. Use this chapter when a problem continues to exist after automatic tuning has been performed and review of the Performance Tools Advisor output has been completed.

Note: It is important to be familiar with the contents of Performance Management Methodology, Guidelines for Interpreting Performance Data, System Performance Tuning Tips, Design and Coding Tips.

The following flowchart is an expansion of stage 3 of the performance management flowchart and shows the steps using Performance Tools output, identifying and making system level changes, and identifying and making application level changes.

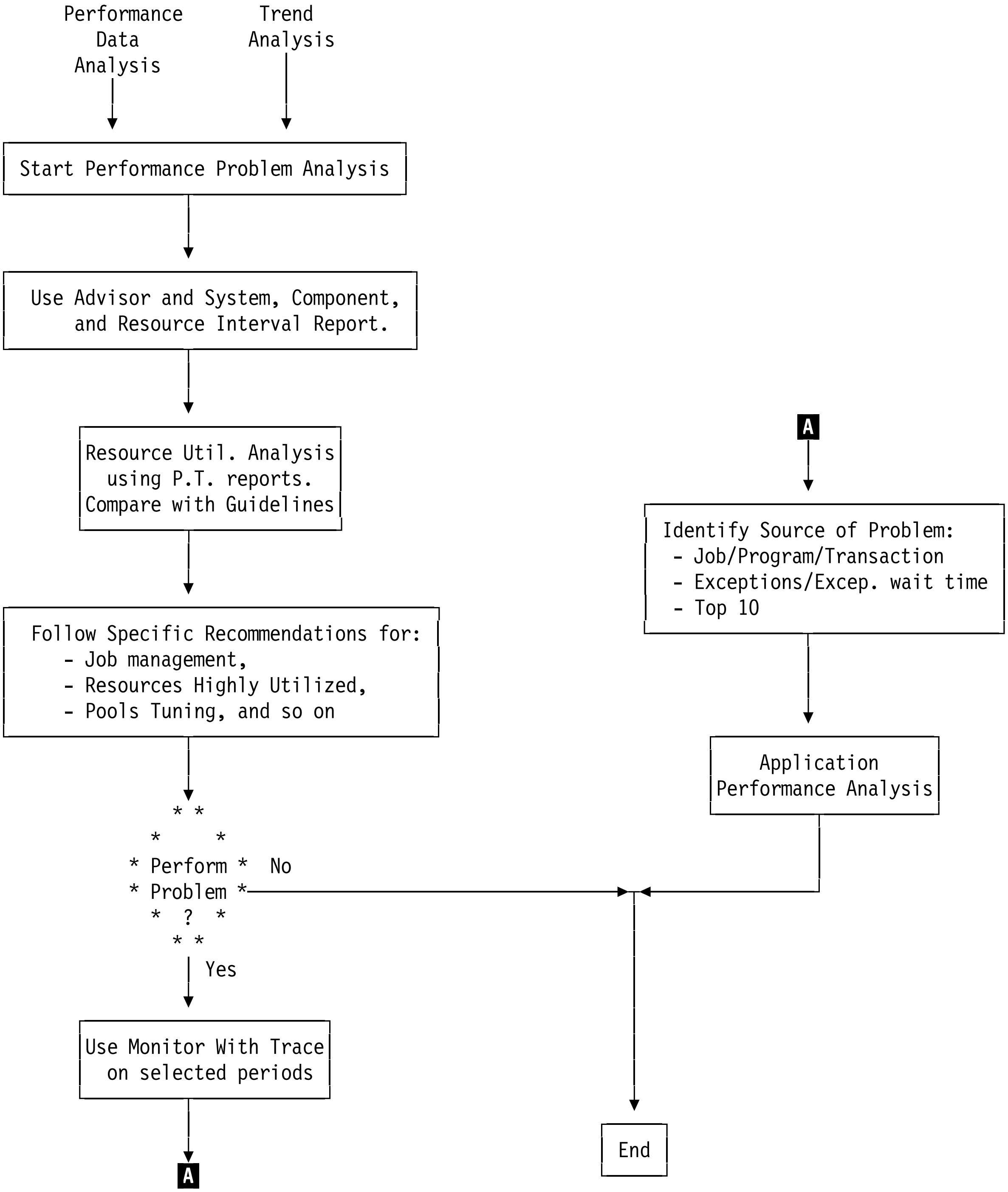

Performance Data Analysis
Trend Analysis
Start Performance Problem Analysis
Use Advisor and System, Component, and Resource Interval Report.
Resource Util. Analysis using P.T. reports. Compare with Guidelines
Follow Specific Recommendations for: - Job management, - Resources Highly Utilized, - Pools Tuning, and so on
Perform Problem ?
No
Yes
Use Monitor With Trace on selected periods
A
A
Identify Source of Problem: - Job/Program/Transaction - Exceptions/Excep. wait time - Top 10
Application Performance Analysis
End

Figure 29. Performance Problem Analysis Flowchart

9.1 The Problem Analysis Cycle

Performance analysis is a method of investigating, measuring, and correcting deficiencies so that system performance meets the user′s expectations. The problem solving approach is:

1. Understand the symptoms of the problem.

2. Use tools to measure and define the problem.

3. Isolate the cause.
4. Correct the problem.
5. Use tools to verify the correction.

Once the apparent cause (or causes) have been isolated, you can propose a solution. The solution can be as simple as tuning the storage pools or a complex one requiring application recoding.

To achieve the optimum performance, you must understand the relationship between critical system resources and attempt to balance the resources that are CPU, main storage, disk, and communication lines. However, any improvement can only come through analyzing the critical resources and contention for system and application objects.

9.2 When To Do Problem Analysis?

The problem analysis cycle can be triggered in two different ways:

- After performance data analysis:
  - When the Performance Tools Advisor is either not used or problems remain after following its recommendations and conclusions
  - From an end user complaint
- From trend analysis that shows some significant changes from previously collected performance data.

Before you start analyzing the data, you should understand under what circumstances the problem has occurred.

Who is Experiencing the Abnormal Situation: It is useful to find out if the situation is affecting only a few end users or most active users on the system. If there are only some end users affected, you should find out what they have in common, such as using the same application, using the same master file, using the same local workstation controller (WSC), the problems are only on a remote communication line versus local or LAN attachment, or problems occur only when many jobs are abnormally terminating at the same time, and so on.

Sometimes a single job can cause severe degradation on many other jobs. This can be a CPU-intensive job that is running at a high priority, for example, a Save/Restore of a library with many objects, or a query of a ten-million record file. Find out if someone ran a job interactively that is normally run in batch. Review the run attributes of the job on the system console.

9.3 Averages versus Peak Workloads

Is the performance always unacceptable or do the users notice degradation regularly? For example, are there problems before and after lunch break or at the end of a working day? If performance is always unacceptable, you should review Chapter 11, “System Performance Tuning Tips” on page 173 and review the system values before continuing the problem analysis. In the case that performance fluctuates regularly, you should start performance data collection (STRPFRMON command) before noticeable degradation. If a poor performance situation is already present, you should use the techniques discussed in Section 9.4, “Reviewing the Measured Data” on page 130.

9.4 Reviewing the Measured Data

There are a number of tools that allow you to interactively review the performance of your system. These are:

WRKSYSSTS

This command gives you a group of statistics that depict the current state of the system. It shows the number of jobs in the system, the disk usage in system ASP, the total amount of disks in the system and the number of addresses used. By selecting assistance level 3 (Advanced), you can monitor all of the memory pools, database and non-database faults, and activity level changes at a glance.

Provided that the subsystems use separate memory pools, you can decide if enough main memory is provided for all of the subsystems. If there is only one pool with a high non-database faulting rate, find out which subsystem uses that pool and monitor that subsystem with the WRKACTJOB command to find out what jobs are active in the subsystem.

Using this display for tuning is described in 198.

WRKACTJOB

This command is used to examine the CPU used and the disk I/O operations done by each job currently active. You can rearrange this display simply by moving the cursor to the desired column and pressing the PF16 key. For example, to find out which job uses the largest amount of CPU, move the cursor to column ″CPU %″ and press the PF16. The display is rearranged to show all the jobs in the system in descending order according to the CPU % used.

You can also find out the amount of synchronous I/O per transaction by first pressing PF11 for more fields and dividing the AuxIO (Disk I/O) by Int (Elapsed Interactions). Information about response time, run priority, and the pool in which the job is run are also displayed. Please note that the result is the **average amount** of I/O during the observation period. Also note that if you do not want to impact the response times of all the other users in the same subsystem, you should lower the RUNPTY (execution priority) of your job through the CHGJOB command to 21.

WRKDSKSTS

This command shows performance and status information about disk units on the system. The column to pay attention to is ″%busy″ located on the right-hand corner of the display. However, this can be off by 25%, more or less. Use it as an indicator to look at the System or Component report. Do not use the values shown for capacity planning. Use the Performance Tools/400 reports and BEST/1 for capacity planning.

You should compare values on this display to Table 33 on page 383 in Appendix A, “Guidelines for Interpreting Performance Data.” Additional information about protection and ASP is achieved by pressing PF11.

WRKSYSACT

The quickest way to analyze a problem situation is to do it interactively through the WRKSYSACT command. This command shows only the jobs that have been **active** during the last observation interval. This command also uses fewer system resources than all of the other commands discussed here.

On small to medium systems, you can use the automatic refresh with a rate of five seconds and see if the same job or task surfaces in many intervals. On larger systems or systems that are busy, an interval of 30 seconds is more appropriate. If the same user keeps appearing on the display, you should go and see what the user is doing.

Go to the user and record everything the user does because that may be the key to solving the problem: the user may bypass an error message by pressing Enter, the user may press incorrect function key, and so on. Find out what application is used and try to re-create the situation yourself. If you do not find anything suspicious or easily correctable, you should find out what programs were used during the period of bad performance. See Section 9.7, "Application Level Problem Analysis" on page 141 for more information.

DSPPFRDTA

This command is a part of the Performance Tools/400 (5716-PT1) licensed program. The command can be used to analyze either real-time data or data collected previously. In both cases, the STRPFRMON command must be issued **before** you can analyze the performance data.

- Summary data:

  To analyze system-wide data with the DSPPFRDTA command, start the performance monitor with parameter TRACE(*NONE).

- Refer to *Performance Tools/400 Version 3*, SC41-4340, for more information about the DSPPFRDTA command.

9.5 System-Wide Performance

The ways to analyze system-wide performance are:

- Interactively, for example, with the WRKSYSSTS command
- Using Performance Tools/400 reports or the Display Performance Data (DSPPFRDTA) command

9.5.1 Memory Usage

The following displays and reports give you information on how effectively memory is being utilized.

WRKSYSSTS: Main memory faults and page rates are displayed in real-time on a per pool basis.

This command provides information on paging rates of all the pools. For guidelines, see Table 34, Table 35, and Table 36 in Appendix A, "Guidelines for Interpreting Performance Data" on page 383.

DSPPFRDTA: The Display Performance Data command provides an interactive display of data given in the System, Component, and Interval reports.

DSPACCGRP/ANZACCGRP: These commands open files, and file I/O counts. Display and Analyze Access Group provides data on the size of the ″currently in use″ part of the PAG. PAG size can be affected by reducing the number of active programs, the number of display and database files open, and the number of display formats and database buffers allocated for the files.

Table 10 provides a cross reference of resource, report, and guidelines for memory usage.

Table 10. Memory Utilization

Resource Description	**Where to Look**	**Compare With**
Machine pool NDB page fault	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Table 34 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Sum of DB and NDB Page Faults for Each Pool	System Report: Storage Pool Utilization, ADVISOR	Table 35 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Sum of DB and NDB Page Faults in All Pools	System Report: Storage Pool Utilization, ADVISOR	Table 36 on page 384 Appendix A, “Guidelines for Interpreting Performance Data.”
Pool Size By Interval	Pool Report: Pool Activity	
The Pool with the Highest Fault Rate for Each Time Interval	Component Report: Component Interval Activity	

9.5.2 CPU Performance

This section reviews reports and displays that can assist in monitoring CPU performance.

WRKACTJOB: This command allows you to determine:

- What is the utilization percentage of CPU?
- How much CPU does each job use, both in terms of percentage and for how long a time total?

If the interactive utilization percentage of CPU is always more than 85, you should try modeling to see if a faster CPU helps. See Table 33 on page 383 in Appendix A, “Guidelines for Interpreting Performance Data” for more information. Please remember that you should not use measurement times that are either under five minutes or over 30 minutes long.

Table 11 and Table 12 on page 133 cross reference various CPU related resources, where to find them, and where to find guidelines for interpretation.

Table 11 (Page 1 of 2). CPU Utilization, Seconds

Resource Description	**Where to Look**	**Compare With**
Interactive CPU	System Report: Resource Utilization Expansion, DSPPFRDTA, Transaction Report	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”

Table 11 (Page 2 of 2). CPU Utilization, Seconds

Resource Description	**Where to Look**	**Compare With**
CPU seconds per transaction	System Report: Resource Utilization, Transaction Report: Job Summary	
CPU Queuing Multiplier	Transaction Report: Job Summary, System Summary Data, System Report	Table 41 on page 386 Appendix A, “Guidelines for Interpreting Performance Data.”
CPU Queuing Multiplier by Job Priority	System Report	
Total CPU usage by job type	System Report: Resource utilization expansion	
Total CPU usage by individual jobs	Component report: Job Workload Activity,Transaction Report	
CPU utilization and seconds per job and system task	Transaction Report: Job Summary, System Summary Data	
CPU Usage by Subsystem and Pool by Interval	Pool Report: Subsystem activity	
Job Maximums of CPU, I/O, Transactions and Response Time by Pool	Pool Report: Subsystem Activity	
CPU Time by Job Per Interval	PRTACTRPT, Component Report	
CPU Time by LIC Task Per Interval	PRTACTRPT, Component Report	
Note: Use the CPU seconds per transaction values in Appendix A as a ″reasonability measure.″ Verify that any job exceeding the values is performing the work required.		

Table 12. Exceptions

Resource Description	**Where to Look**	**Compare With**
Authority Lookup	Component Report: Exception Occurrence Summary	Appendix B, “Program Exceptions.”
Size (Arithmetic Overflow and Binary Overflow)	Component Report: Exception Occurrence Summary	Appendix B, “Program Exceptions.”
Verify	Component Report: Exception Occurrence Summary	Appendix B, “Program Exceptions.”
Decimal Data	Component Report: Exception Occurrence Summary	Appendix B, “Program Exceptions.”

9.5.3 Disk Performance

In this section, we direct you to various reports and other references to assist you in reviewing disk performance.

9.5.3.1 Disk Performance Reports/Displays

There are a number of standard Performance Tools/400 Reports and AS/400 CL commands that can give you useful disk information.

System Report:

- Shows disk I/O by job type (Batch, System, Interactive, Pass-through, and so on).
- Shows IOP utilization.
- Shows ASP, checksum set, and mirrored units.
- Shows the disk unit size.
- Shows the I/O rate.
- Shows disk IOP and device service time.

Component Report:

- Shows synchronous and asynchronous disk I/O per second, by interval.
- Shows summary of highest used device in the interval.
- Shows synchronous and asynchronous disk I/O per job total.
- Shows summary of database journal deposits (entries), bundle (blocks of deposits) writes for both user journaling and, for System Managed Access Path Protection (SMAPP) support, system access path journal deposits and bundle writes and access path recovery time estimates.
- Shows all selected intervals, utilization, size, number of overruns and underruns, and seek activity by unit.

Transaction Report - Summary Report:

- Shows synchronous and asynchronous disk I/O per transaction per job.

WRKSYSSTS: Shows total disk space available and in use.

WRKDSKSTS: Shows disk reads and writes per drive and the disk utilization.

WRKJOB: Shows the number of disk I/O operations by file name.

WRKACTJOB: Shows the number of disk I/O operations by job.

WRKSYSACT: Shows the number of disk I/O operations by job and LIC task. These are further separated as synchronous and asynchronous operations.

9.5.3.2 Information about Disk Performance

Table 13 (Page 1 of 2). Disk Utilization

Resource Description	**Where to Look**	**Compare With**
Disk Arm Utilization	System Report: disk utilization, WRKDSKSTS	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”

Table 13 (Page 2 of 2). Disk Utilization

Resource Description	**Where to Look**	**Compare With**
Disk IOP Utilization	Component Report: IOP utilization	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Disk Physical I/O per Transaction (Average)	System Report: resource utilization, transaction report	
Disk Physical I/O per Transaction per Job.	Transaction Report: job summary	
Synchronous and asynchronous DB and NDB I/O per job by interval	Job interval report	
Sync and async disk I/O per job or LIC task per interval	PRTACTRPT	
Database journal deposits and bundle writes to user and system (SMAPP) journals	Component Report: database journal summary	
Sync and async Disk I/O by subsystems and pools by interval	Pool Report: subsystem activity	
Note: Any asynchronous disk I/O performed by the system QDBSRVnn jobs on behalf of a user job are not included in the job′s asynchronous I/O totals shown on performance reports. See Section 11.1.7, “Database Server Jobs (QDBSRV01... QDBSRVnn)” on page 175 for more information.		

9.5.3.3 Other Useful Reports

And finally, here are a few other places to look for disk related performance information.

Resource Report: Shows by interval the disk I/O per second, reads and writes per second, average amount of data transferred per disk I/O, highest utilization and service time disk unit, and total space used.

Shows by unit and interval the unit identification data (bus, IOP, ASP), reads and writes per second, average data transfer size, unit and IOP service time average, and average device I/O queue depth.

Pool Report: Shows the highest number of disk I/O operations by a job running in a pool during an interval.

9.5.4 Local Workstation IOP and Communications Lines

Local workstation IOPs and communications lines can have a considerable impact on interactive performance. In many instances, poor client/server performance can be traced to communication line problems. Here are some reports that help you in ascertaining if you have a workstation IOP or communications line problem.

9.5.4.1 Local Workstation IOP and Multi-function IOP Performance Data

Performance Tools: The Component Report shows local workstation IOP and multifunction IOP utilizations. For local workstation IOPs, twinaxial utilization is also shown.

It is possible to have either high local workstation IOP utilization and low twinaxial utilization **or** low local workstation IOP utilization and high twinaxial utilization. High IOP utilization can occur if there is heavy use of the text assist functions for an OV/400 editor. High twinaxial utilization can occur if there is a significant amount of high-speed printer output or Client Access/400 shared folder or file transfer work going on.

9.5.4.2 Line Performance Data

Performance Monitor: The STRPFRMON command can optionally collect remote response time data from 5494 remote controllers with microcode release 1.1 or later installed on the 5494. Communication IOP and line performance data is always collected. There are a number of monitor database files used to collect this data.

Performance Tools: The System Report Communications Summary shows average and peak line utilization over the report period. The Resource Interval report shows communication line details per time interval selected. The customer is free to use a query product or user-programming to develop unique reports not available with Performance Tools.

Note: Performance Tools also provides graphic support for line utilization and can show up to 16 lines on a single graph page.

QSYSOPR, QHST Messages: Error failure, threshold, and communication job start and end messages can be in this message queue or the history log.

Communication Error Log: Communication errors are logged in the system error log regardless of Performance Monitor activity. Each entry is time stamped. Use the STRSST command to view the logged data. Assistance from IBM service in interpreting the log data is needed in most cases.

9.5.4.3 Information about Local Workstation and Communications Performance

Table 14 (Page 1 of 2). Lines and IOPs Utilization

Resource Description	Where to Look	Compare With
Local WS IOP	Component Report: IOP utilization	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Multifunction IOP	Component Report: IOP utilization	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Communication IOP	Component Report: IOP utilization	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”

Table 14 (Page 2 of 2). Lines and IOPs Utilization

Resource Description	Where to Look	Compare With
File Server IOP	Component Report: IOP utilization	IOP reported is the one for exchanging data between the Integrated PC Server and AS/400 disk storage. No guideline available at this time. Attached LAN lines are reported under remote lines, LAN lines. See Section 13.5.8, “Integrated PC Server Performance Monitor Data Queries” on page 354 for information on how to obtain cache read/write hit rates and 486 CPU utilization percentage guidelines.
Remote Lines, LAN Lines	System Report: Communication Summary, Resource Report	Table 33 on page 383 Appendix A, “Guidelines for Interpreting Performance Data.”
Communications I/O Count by Job Type	System Report: Resource utilization	
Line Utilization and Activity (input/output)	System Report: Communications summary	
Communications Gets and Puts per Transaction by Job type	System Report: resource utilization expansion	
Communication I/O Per Job	Component Report: job workload activity	
Local and Remote Workstation Response Time Distribution	Component Report: local workstations - response time buckets	
Local and Remote Workstation Response Time Distribution By Interval	Resource Report: local workstation IOP utilization and remote workstation response times	

9.5.4.4 Other Useful Reports/Displays

Resource report:

- Additional line utilization data by interval
- Response time counts per ″response time period″ for local workstations and optionally for remote 5494-attached workstations

Query:

- The Performance Tools reports do not include all data or they show certain combinations of data. A common use of a query is to tie together more complex analysis structures such as jobs, pools, lines, and so on.

See Section 11.12, “Communication Performance Considerations” on page 209 for information on the OS/400 Performance Monitor communications related database files.

Display Performance Data (DSPPFRDTA):

- Provides a combination of system, component, and resource report information.

Work with System Activity:

- LIC communication task activity (CPU, disk I/O, frequency).

System Service Tools:

- Communications and device error logs provide information about permanent errors and threshold readouts.

9.5.5 Activity Level Performance Data

Performance Monitor, WRKSYSSTS: The OS/400 Performance Monitor trace, sample data, and WRKSYSSTS provide information on activity level usage and job state changes.

9.5.6 Activity Level Performance Reports/Displays

System Report: Job state changes (movement in and out of activity level) per pool for the total collection period.

9.5.7 Comparing with Activity Level Guidelines

Table 15. Activity Level

Resource Description	Where to Look	Compare With
Activity Level for *BASE and spooled writer pool	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Figure 3-8, Figure 3-9, and Figure 3-10 on page 3-9 and page 3-10 in *Performance Tools/400 Version 3*, SC41-4340.
QINTER Activity Level	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Figure 3-11 on page 3-10 in *Performance Tools/400 Version 3*, SC41-4340.

9.5.8 W – I and A – W Ratio Guidelines

More simply, get the activity level to where W – I is a little bit above zero and increase the activity level by 2.

If W – I is always zero, the activity level is too high.

9.5.9 Comparing W-I and A-W Ratio Guidelines

Table 16. W-I and A-W Ratio

Resource Description	Where to Look	Compare With
W-I/A-W	System Report: Storage Pool Utilization, WRKSYSSTS	Table 37 on page 384 Appendix A, “Guidelines for Interpreting Performance Data.”

9.5.10 Other Useful Reports

Component Report: Shows job state changes by pool summarized over selected time intervals.

Pool Report: Shows job State changes by subsystem and pool for each selected time interval.

Shows pool activity level for each interval. It may change during the time period due to operator action, an OEM automatic tuner, or the Version 2 system automatic tuning through QPFRADJ. The value shown is the value at the time of the sample.

Display Performance Data: Provides online display of system, component, and pool interval report data.

9.5.11 Advisor

The Advisor analyzes Performance Monitor data *after data collection has completed*. All or a subset of the time intervals can be selected for analysis. The Advisor can be run interactively or in batch when the Submit Job command contains the Analyze Performance Data (ANZPFRDTA) command in the CMD parameter. The output of the Advisor is a set of recommendations and conclusions regarding:

- CPU utilization of ″high priority″ (20 or lower value) jobs
- Performance analysis of interactive transactions by using the trace data collected with Performance Monitor
- Main storage utilization and Wait-To-Ineligible versus Active-To-Wait ratio; this addresses page faulting and activity level analysis.
- Disk utilization and other disk activity
- IOP utilization
- CPU utilization as a result of authority lookups
- Impact on CPU utilization as a result of exceptions (such as numeric overflow and verify)
- Communication line utilization and error percentages

The Advisor output is grouped under the following headings:

- Recommendations
- Conclusions
- Interval conclusions

The Advisor suggests changes to pool sizes and activity levels. These changes are not made dynamically, but only after the operator tells the Advisor to make the changes or to ignore the recommendations. The tuning is done by using the F9 key on the Display Recommendations Display. Pool and activity level changes can be made to all pools on the system, including system, shared, and user-defined. Since the Advisor can analyze performance data collected from other systems but restored on the system for analysis, the changes are not made if the data is collected on another system.

The Advisor also tells you what report you should run to get more information for further problem analysis. See ″Chapter 5. Advisor″ in *Performance Tools/400 Version 3*, SC41-4340 for detailed information.

9.5.12 Remote Workstation Response Times

If you have 5494 workstation controllers included in the data collection, you have information about:

- The number of active workstations on each controller
- The range of response times for the workstations
- The average response time for the workstations

The Performance Tools/400 System Report and Resource Report list this 5494 response time data. Note that you must specify a STRPFRMON RRSPTIME value other than the default *NONE to have the response time data collected from the active 5494 controllers. A value of *SYS uses the same response time slots (0-1 seconds, and so on) as for the LRSPTIME (local response time) parameter.

9.6 User Level Problem Analysis

Are all the users affected by poor performance or is there only a small easily defined group of users affected? What do these users have in common? Do they use the same application; are they sharing the same (possibly small) memory pool? Is there only one user suffering from poor performance? How does this user differ from the rest of the users? The first step to determine a user-level problem is identifying the affected user or users and after answering these questions, the solution is much closer. For one way of identifying a user level problem, see Section 9.4, “Reviewing the Measured Data” on page 130.

User level problem analysis is done:

- By WRKACTJOB command
- By WRKSYSACT command
- By DSPPFRDTA command
- By analyzing Performance Tools/400 reports

9.6.1 Print Job Summary Report

This is the starting point for user-level problem analysis. Find out if the user appears in the ″Job Statistics″ section of the ″Job Summary Report″. Next look at the ″Individual Transaction Statistics″ section to see what program that user is using. Is this user the only one using this program? If no, is this user the only one with a performance problem? If all the users of this program have problems with performance, you should see Section 9.7, “Application Level Problem Analysis” on page 141 for more information.

Refer to the ″System Summary Data″ section of the report, ″Analysis by interactive Response Time″ to see how your response time objectives are met.

9.6.2 Print Transaction Summary Report

The Transaction Summary report provides you information about response times, CPU utilization, and disk I/O by job. This report can be used for both user-level problem analysis and application-level problem analysis.

If the Job Summary section shows jobs that have high response times, high disk I/O activity, or high CPU utilization, use the Transaction Detail report to investigate further. However, you should always print the Summary report first because the Transaction Detail report and the Transition report provide detailed information. By using the Summary report, you can choose to print only the intervals or users that have performance problems instead of printing thousands of pages of irrelevant data.

9.7 Application Level Problem Analysis

Is there a problem with one application only? Are there only some operations that are slow?

Application level analysis is based on Performance Tools/400 reports. See the ″Interactive program statistics″ section of ″Job Report″ for the top 10 programs with highest resource utilization such as:

- CPU per transaction
- Disk I/O per transaction
- Response time per transaction
- Database reads/writes per transaction
- Non-database reads/writes per transaction

Compare this information to the averages shown in the System Report and determine if their difference warrants further investigation.

9.7.1 Print Transaction Summary Report

From the Transaction report, you can select those programs that show a frequent high resource utilization. These programs should be analyzed in deeper detail using the other tools listed in Section 9.7.2.1, “Programmer Performance Utilities” to find out the cause of the problem.

9.7.2 Useful Tools for Performance Analysis

Here are some useful tools to help you find problems at the application level.

9.7.2.1 Programmer Performance Utilities

Job Trace: OS/400 commands such as STRSRVJOB, ENDSRVJOB, and TRCJOB can be used to produce trace job information. The Performance Tools commands such as STRJOBTRC, ENDJOBTRC, and PRTTRCSUM (now found in IBMLIB) can also be used to produce trace job information.

The PRTTRCSUM command (now found in IBMLIB) is a productive tool for finding problems.

The trace job output can be used to determine the following information to analyze job performance:

- Programs called and calling sequence and frequency
- Wall clock time of the program call and return sequence
- CPU time used by each program
- The number of synchronous DB and NDB disk I/Os per program called
- The number of full and shared file opens
- Messages received by each program

Do not use the wall clock time (TIME heading) or CPU time (CPU TIME heading) to estimate the actual time used by each program. The implementation of trace job inflates the real values to those shown in the trace job data. However, you can use the time values to identify relative differences among the programs listed.

Be aware, for a job with hundreds of user program/procedure calls, the trace job may have a significant impact on CPU utilization.

If you do not have access to IBMLIB, a similar report can be produced with the Performance Explorer *STATS option using the DTAORG(*HIER) parameter when defining the PEX definition.

Performance Explorer *PROFILE option: The Performance Explorer *PROFILE option is a replacement for the Sample Address Monitor (SAM) that was available as part of Performance Tools/400 prior to V3R6.

Once a program has been identified as being a high CPU user, it is possible using PEX *PROFILE to determine within the program (down to the instruction level) where the CPU is spending most of its time. Thus, you can performance tune at the instruction level of an individual program. This is especially useful where many users run a copy of the same program and small improvements are replicated many times.

The following technique is common to all three Performance Explorer options (*PROFILE, *STATS, *TRACE) and may be used to collect and analyze PEX *PROFILE data.

The Performance Explorer work cycle is made up of three tasks:

1. The first task in the cycle is to define the manner in which performance data is to be collected. This is done by defining the data to be collected using the Add Performance Explorer Definition (ADDPEXDFN) command. After the definition is completed and saved, you are ready to continue to the second task in the cycle of work.

2. The second task is collecting the performance data. This creates a data file containing the specified performance data according to the definition created in the first task.

   To start collecting performance data, use the Start Performance Explorer (STRPEX) command. You can specify whether to start a new performance explorer session or resume an already active session.

   To end the performance explorer session, use the End Performance Explorer (ENDPEX) command. An option on this command allows you to suspend a collection temporarily so that you can resume the collection at a later time (using the STRPEX command).

3. The third task is to analyze the performance data. This assists us in understanding problem areas where performance can be improved. To analyze the data, you must create performance explorer reports using the Print Performance Explorer Report (PRTPEXRPT) command.

To delete performance explorer data, use the Delete Performance Explorer Data (DLTPEXDTA) command. This discards performance data from a set of database files.

For a more detailed description on using PEX, see the redbook *AS/400 Performance Explorer: Tips and Techniques*, SG24-4781. Limited information can be found in *Performance Tools/400 Version 3*, SC41-4340.

STRDSKCOL and PRTDSKRPT: The STRDSKCOL command and PRTDSKRPT command are not available on OS/400 V3R6. It may be possible to derive similar information by using a query over data obtained from PEX *TRACE option. This is a non-trivial exercise.

DSPACCGRP and ANZACCGRP: Analyze Process Access Group activity with these commands. Collect the data with the DSPACCGRP command and direct the output to a database file. The command lets you select jobs by generic job or user name, or by type (interactive or all).

Use ANZACCGRP to print a summary of the data in the file. For each job type, it shows:

1. How many jobs exist
2. The number of files that are in use in each job and the amount of I/O done by the job
3. What files are open in the system, what duplicate files a job may have, and the amount of I/O going on for each file

Analyze Job PAGs to see if savings can be made. Opening and closing seldom used files each time they are used saves buffer space. In some cases, display files have many formats but a job uses only one or two. Placing these formats into a separate display file (for example, based on application function) can reduce PAG size. This reduces the number of disk I/O operations to read and write the PAG, and saves space while the PAG is in memory. This is valuable on a system with limited main storage.

Note: Beginning with V2R3, DSPACCGRP and ANZACCGRP no longer provide complete program storage information. At this time, there is no plan to provide this information. File open information and the number of active programs are still provided.

Beginning with V2R3, the internal job structure was significantly modified to support the Integrated Language Environment (ILE) as ILE C/400 was introduced in this release.

The biggest changes are that the Program Automatic Storage Area (PASA) and Program Static Storage Area (PSSA) now exist only as ″named spaces″ (no data). This enables Original Program Model programs to work without impact. Program static information and variable information are stored and managed in several different storage areas and there is no function that can associate all of the storage to a specific program, module, or procedure. The LIC Process Control Block (PCB) and the OS/400 Work Control Block (WCB) point to these other control areas. The Data Management queue (file open and file override information) implementation still provides information that can be used to analyze the job′s efficient use of file opens and closes.

Initially, there is a Process Access Group (PAG) allocation for the PCB, WCB, and a Process Activation Work Area (PAWA). The PAWA contains pointers to the Activation Control which is the static, auto, and heap storage carved up into smaller work areas to reduce the working set size of the job. The working set size is the portion of the PAG that is needed for current processing and there are new algorithms to calculate and minimize the size of the working set.

In V3R6, the Work Control Block Table implementation was modified from a single table to multiple tables, each 1KB in size. This was done to improve access to the various call stack/program heap storage areas.

9.7.3 Print Transaction Detail Report

If you need a more detailed problem analysis, print a Transaction Detail report by specifying RPTTYPE(*TNSACT) on the PRTTNSRPT command. The Transaction report output has two parts:

- The details, which show data about every transaction in the job.
- The summary, which shows data about the overall job operation.

If there are response times that are not acceptable compared to objectives, you should read the report further.

The next section you should see is the Job Summary Data and especially the Synchronous Disk I/O Counts. If there are, for example, 200 DB Reads (database read operations) per transaction, the response times are unacceptable.

9.7.4 Print Transition Report

If you want to know all of the state changes within a transaction, run the Transaction report by specifying RPTTYPE(*TRSIT) on the PRTTNSRPT command. Remember to use the select/omit parameters or you receive several thousand pages of printout.

The Transition report is composed of two sections:

- Transition Detail, which shows each state transition made by the job (for example, active-to-ineligible and transaction boundaries). For more information about transaction boundaries, see the ***Field Description and Sample Reports***.
- Summary, which shows the same data as the summary output from the Transaction report.

You may see in the Transaction report - Seize/Lock Conflict reports that object ″ADDR 00000E00 0002IUSE″ is being held for a relatively long time. This refers to the internal object ″Database File in Use Table″. This indicates frequent occurrences of one of the following conditions:

- File opens/closes
- File creates/deletes
- Clear physical file member
- Reorganize physical file member, and so on

Since these functions have a significant impact on system and job performance, you should try and reduce their usage.

You may also see the I/O Transaction boundaries in the Transaction report. They indicate the trace point such as:

- SOTn:
  - Start of a transaction

  - Start of the response time for that transaction
  - n = transaction type (1 for Display I/O, 2 for Data queue)
- EORn:
  - End of response time for the transaction
- EOTn:
  - End of resource usage time
  - End of the transaction

See *Performance Tools/400 Version 3*, SC41-4340, for more details.

Chapter 10. Additional Performance Tools

This chapter introduces some additional performance tools that are useful when monitoring performance.

10.1 Performance Tools/400 Agent (5716-PT1)

The Performance Tools/400 Agent feature of 5716-PT1 provides a subset of the full Manager feature of the Performance Tools/400. The Agent is intended for locations where there is no resident AS/400 performance expertise and full Manager functions are not required.

The functions supported by the Performance Tools/400 Agent are:

- Collect Performance Data
- Delete Performance Data
- Copy Performance Data
- Convert Performance Data
- Display Performance Data
- Work with Historical Data
- Advisor:
  - All functions trace data analysis
  - Create, delete BEST/1 Model

A detailed comparison of the functions of the two products is available in appendixes of the *Performance Tools Guides*.

10.2 Performance Management/400 (PM/400)

Performance Management/400 is both a software product (program 5716-PM1) and a service offering. With PM/400 job scheduler, support is provided to automatically start and end the OS/400 Performance Monitor, summarize the monitor data, and weekly transmit the summarized data to a central site for analysis. The summarized data is used to produce reports and graphs that demonstrate trends in system resource utilizations, interactive transaction rates (throughput), and interactive response time.

The reports and graphs are generated at a central location normally operated by IBM. In a multiple system network, PM/400 can be configured to transmit the summarized data to either a central system with PM/400 installed or to an IBM location. The central system is configured to forward received PM/400 summary data from multiple AS/400 sites to an IBM location for analysis.

In addition to the performance data analysis, the full PM/400 service provides a set of ″historical performance data″ commands that permit a rich set of functions in viewing collected performance monitor data.

Also available as an option to PM/400 is ***″AS/400 Performance Edge″ PE/400***. Performance Edge combines a PM/400-Subset with the AS/400 Service Director (automatically analyzes and reports problems to IBM Hardware Service) and the

AS/400 Forum (provides a bulletin board for AS/400 users that includes access to summarized PM/400 trending data).

Refer to the *Performance Management/400 Offerings and Services, including Performance Management/400 - Subset*, SC41-4347-00, for a complete description of PM/400 support.

Figure 30 shows an example of PM/400 performance data collection and flow. The example shows the U.S. Competency Center in Rochester, Minnesota, U.S.A. as the receiver of the collected data. This is where the data is analyzed, and the reports and graphs are generated.

The PM/400 service offering is supported in several other countries and under consideration in an additional set of countries. In all supported countries, the appropriate APPC configuration, telephone number, and modems must be configured correctly. Contact your local IBM representative to determine country-specific support.

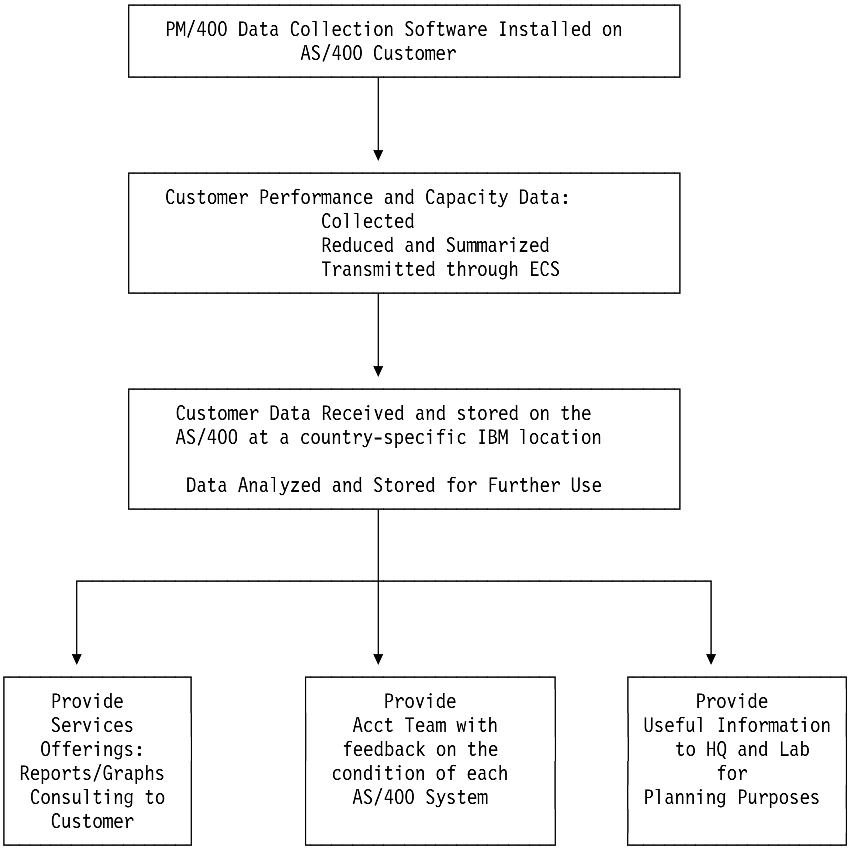

PM/400 Data Collection Software Installed on AS/400 Customer
Customer Performance and Capacity Data:
Collected
Reduced and Summarized
Transmitted through ECS
Customer Data Received and stored on the AS/400 at a country-specific IBM location
Data Analyzed and Stored for Further Use
Provide Services Offerings: Reports/Graphs Consulting to Customer
Provide Acct Team with feedback on the condition of each AS/400 System
Provide Useful Information to HQ and Lab for Planning Purposes

Figure 30. PM/400 Information Flow

PM/400 is not a tool; it is a service offering and a performance methodology that is currently available in selected countries. It assists the AS/400 user in managing the AS/400 resources and in providing consistent service levels.

PM/400 implements and automates a performance methodology by:

- Continuously collecting performance data
- Automatically managing the performance data

- Daily reducing and summarizing the data
- Providing management level reports and graphs that include trend information
- Providing the ″Work with History″ commands that allow a user to manage and diagnose performance problems

10.2.1 PM/400 Levels of Support

If an AS/400 system is set up to run PM/400 and transmit the PM/400 performance summary data to an IBM location, the data is analyzed every month. Each country may implement some or all of the following levels of PM/400 support:

- Notification letters:

  A letter is sent to the customer when a system resource (CPU, memory, DASD) is predicted to exceed the guideline within six months. A second letter is sent as a follow up within two months of the predicted guideline date.

- PM/400 - Subset Reports and Graphs (Performance Edge/400):

  AS/400 Performance Edge is an enhancement to the IBM Maintenance Agreement. It combines standard maintenance services with the following electronic support options at no extra charge:

  – AS/400 Service Director: Automatically reports problems to IBM Hardware Service.

  – PM/400 - Subset: Automates the collection, analysis, and reporting of performance data. The customer receives trend analysis data semi-annually.

  – AS/400 Forum: Provides bulletin board access and has been enhanced to provide summarized trending data from PM/400.

  Although all countries may not offer AS/400 Performance Edge, most countries offer one or more of these services, such as the PM/400 subset.

  The PM/400 - Subset gives the user a subset of the PM/400 Service Offerings reports and graphs semi-annually. Trending information and graphs are provided that show weekly averages.

  Different countries may have a different implementation of PM/400 - Subset. A common practice in most countries is to provide two different levels of service for PM/400 - Subset, depending on whether the customer has an SSA (Software Service Amendment) or MA (Maintenance Agreement) only.

Table 17. PM/400 Report Availability

Report Types	PM/400 Service Offering: Monthly/Quarterly/Annual	PM/400 - Subset Semi-Annual
Management Summary	Yes	Yes
Application Summary	Yes	No
Technical Summary	Yes	Yes
IOP/Communication Summary	Yes	No
Workload Summary	Yes	Yes

Table 18. PM/400 Monthly Graph Availability

Graph Types	PM/400 Service Offering: Monthly/Quarterly/Annual (Color)	PM/400 - Subset Semi-Annual (Black/White)
CPU Utilization by Date	Yes	No
CPU Utilization by Time of Day	Yes	Yes
Response Time by Day	Yes	No
Response Time by Time of Day	Yes	No
Throughput by Day	Yes	No
Throughput by Time of Day	Yes	No
Faulting Rates by Date	Yes	No
Faulting Rates by Time of Day	Yes	No
Disk Space by Date	Yes	No
Disk Arm Utilization by Date	Yes	No
Number of Jobs by Date	Yes	No
Pages Spooled by Date	Yes	No

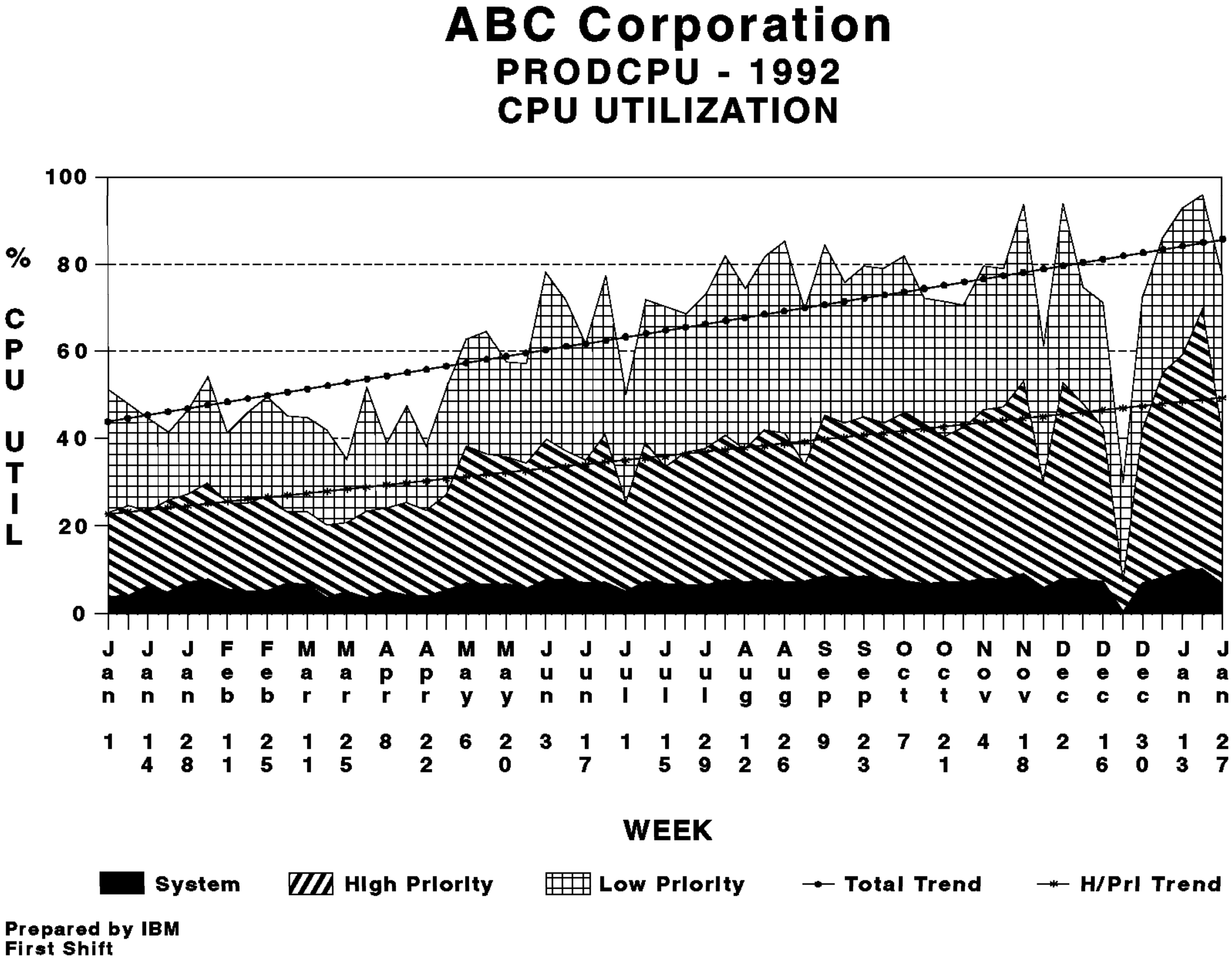


Figure 31. PM/400-PE/400 Sample Graph

- PM/400 - Full Service Reports and Graphs:

  The full service output includes black and white reports and colored graphs.

The graphs include information by day of the month and time of day. This level of detail enables evaluation of workload and job scheduling by day.

This full service includes the PM/400 ″work with historical day″ commands described later in this section.

Summary reports and color graphs with a one month, quarter, or annual trend line produced by IBM are mailed to you monthly, quarterly, or annually, depending on the fee offering. The information provided in the reports and graphs is summarized in the following tables.

Table 17 on page 149 shows available report output for the full service and the subset of PM/400. Table 18 on page 150 shows graphic output for the full service and subset PM/400 options. Figure 32 shows another example of the graphics available.

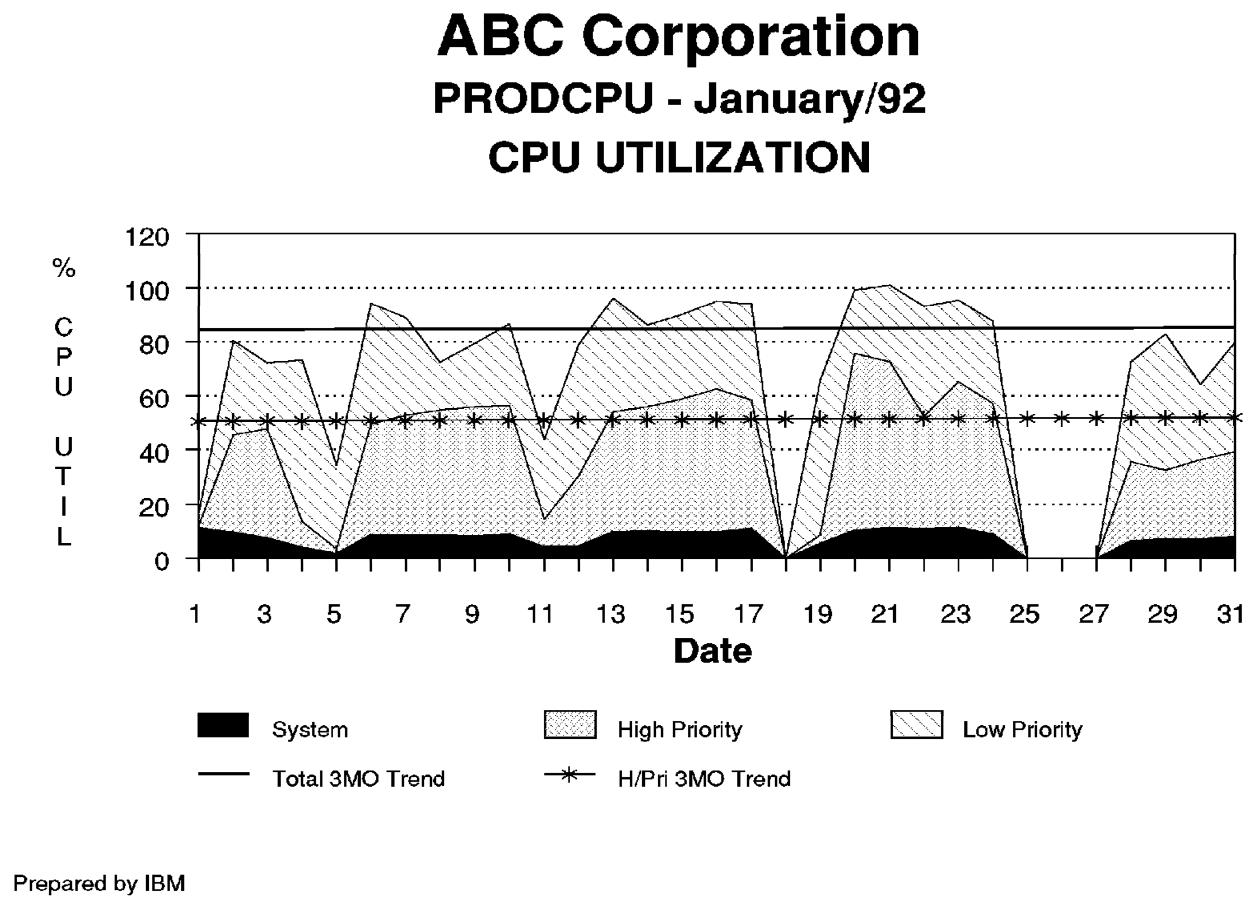


Figure 32. PM/400 Sample Chart

10.2.2 Help Desk Service

In addition to generating periodic performance analysis reports, PM/400 gives you access to a Performance Help desk. This service lets customers who want to free analysts for other tasks, or customers who do not have extensive skills in performance and capacity management, to take advantage of IBM′s own skilled specialists. A toll-free number connects you to an IBM specialist who can make tuning recommendations to improve system performance.

PM/400 complements existing support services available in the United States such as AS/400 Performance Edge/400.

10.2.3 How to Get Started

For U.S. customers, there is a toll-free number (1-800-IBM-4260) where you can get a PM/400 contract or more information about this product.

There are three fee options:

1. Monthly service includes:
   - The monthly summary reports and colored graphs
   - Five additional quarterly summary graphs in March, June, and September
   - Five additional annual summary graphs in December
2. Quarterly service includes:
   - The monthly summary reports and colored graphs for the months of March, June, September, and December
   - Five additional quarterly summary graphs in March, June, and September
   - Five additional annual summary graphs in December
3. Annual service includes:
   - Full set of monthly summary reports and colored graphs for December
   - Five annual summary graphs in December
4. Performance Help Desk:
   - Available to all customers with a PM/400 contract
   - Service billed by the hour for elapsed service time

10.2.4 New for OS/400 V3R6

1. PM/400 is now a licensed program (5716-PM1).
2. There is a new customization display:
   - To set the PM/400 phone number
   - To set the Advantis connection
3. There are new displays:
   - To display the status of PM/400 functions
   - To review a three-month history of the system′s trend information

PM/400, 5716-PM1

Prior to V3R6, PM/400 was a PRPQ (5799-MPG). For V3R6, PM/400 is a no-charge Licensed Program Product (LPP) 5716-PM1 that is shipped with OS/400.

For V3R6, PM/400 has improved ease-of-use by allowing users to enter PM/400 phone numbers on a display and to show the status of PM/400 functions. As a licensed program product, it can be displayed, installed, and deleted using the Work with Licensed Programs menu. And it can be automatically refreshed in the future during an OS/400 software upgrade. This helps to eliminate potential problems that occur when the PM/400 PRPQ code is not upgraded when OS/400 is upgraded, as in previous releases.

V3R6 PM/400 is pre-installed on new AS/400 Advanced Series PowerPC systems shipped from the plant.

Customers who have a previous release of PM/400 installed (that is, as long as they have QMPGLIB on their systems) are notified on the Install Licensed Program Menu during the upgrade to V3R6 to install 5716-PM1. If they are using Automatic Install mode, 5716-PM1 is automatically installed.

New Customization Display

PM/400 provides a new customization display to set the PM/400 phone number instead of having to use the CHGDTAARA command to set up the phone number.

New displays

- New display to show status of PM/400 functions:

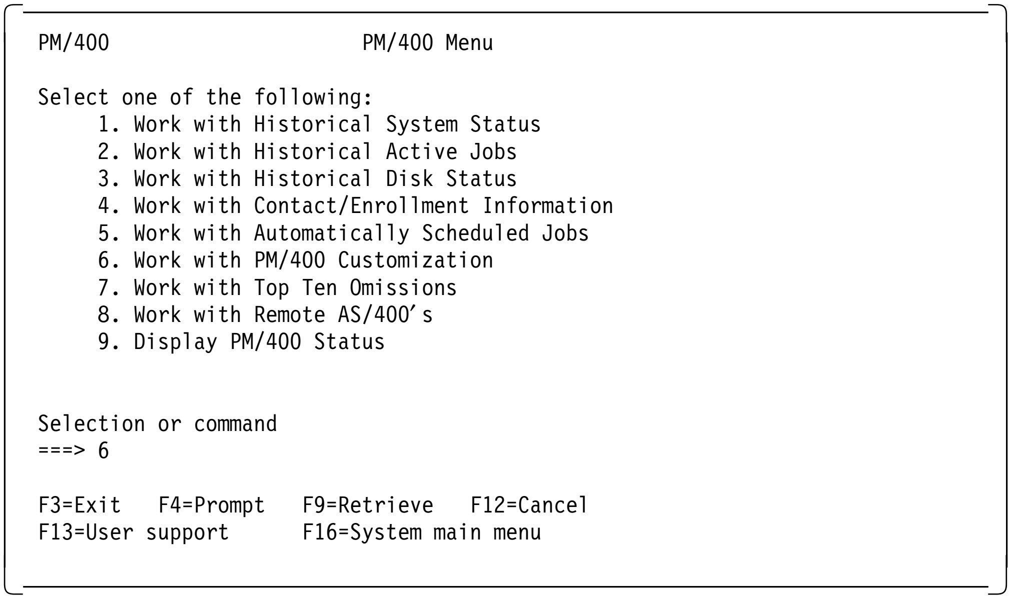
  The status of PM/400 functions is now shown on one display using only one option instead of the many different commands and key strokes that were needed in previous releases.

- New display to review a three-month history of system′s trend information:

  All PM/400 customers are now able to review a three-month history of the system′s trend information without having to wait for the next report from IBM. This assumes that the customer regularly collects and transmits data to IBM (usually on weekly basis).

10.2.5 PM/400 Customization

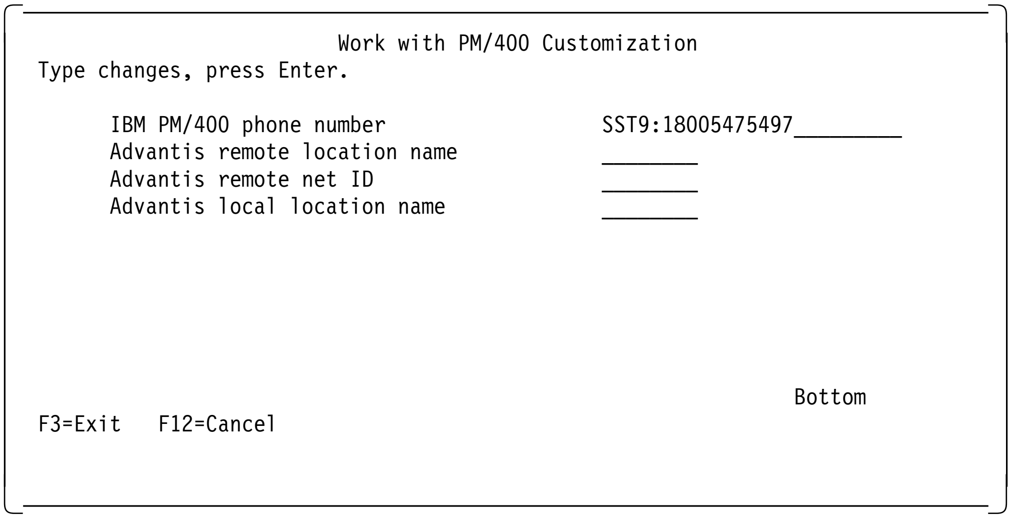
To reach the PM/400 Customization display, take Option 6 on the PM/400 Menu in Figure 33.



```
PM/400                       PM/400 Menu

Select one of the following:
     1. Work with Historical System Status
     2. Work with Historical Active Jobs
     3. Work with Historical Disk Status
     4. Work with Contact/Enrollment Information
     5. Work with Automatically Scheduled Jobs
     6. Work with PM/400 Customization
     7. Work with Top Ten Omissions
     8. Work with Remote AS/400′s
     9. Display PM/400 Status


Selection or command
===> 6

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=User support      F16=System main menu
```

Figure 33. PM/400 Menu - Display 1

The Work with PM/400 Customization function provides the user with the ability to establish global parameters for the operation of the PM/400 software. Customization includes setting the priority limit, trend and shift schedules, performance data library, and purge specifications. In addition, the PM/400 customization allows the user to specify the PM/400 data number or set up PM/400 to use Advantis (IGN).



```
                         Work with PM/400 Customization
 Type changes, press Enter.

       IBM PM/400 phone number                SST9:18005475497_________
       Advantis remote location name          ________
       Advantis remote net ID                 ________
       Advantis local location name           ________







                                                               Bottom
 F3=Exit   F12=Cancel
```

Figure 34. Work with PM/400 Customization - Display 2

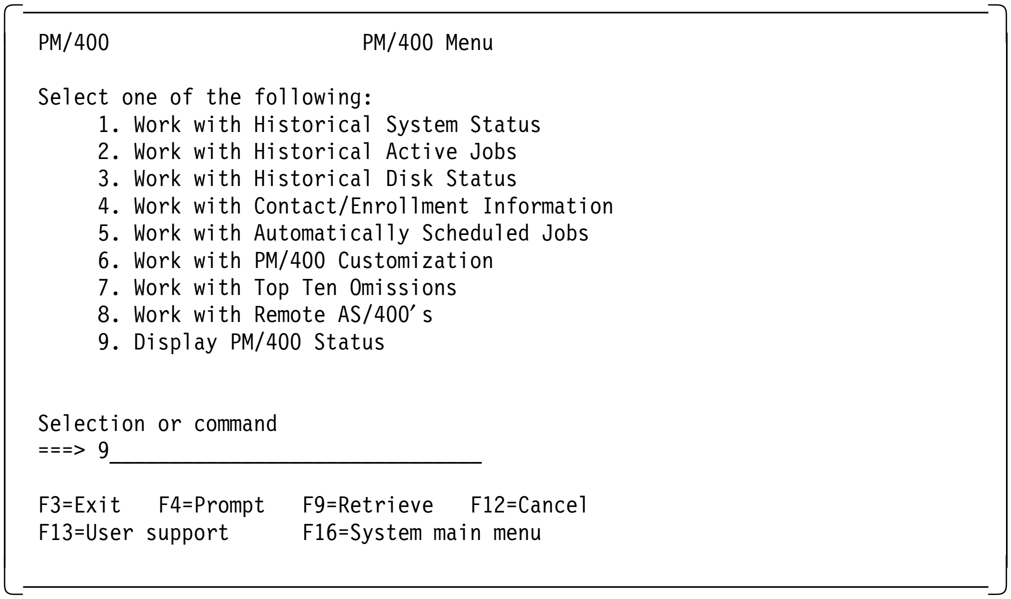
From the Command Entry display, you can start the Work with PM/400 Customization function by typing the following command at the command line and pressing the Enter key:

```
GO QMPGLIB/PM400
```

Type **6** on the command line of the PM/400 Menu (see Display 1) and press the Enter key. The Work with PM/400 Customization display is shown. The second Customization display allows you to set up the PM/400 communications (see Display 2). You can either modify the PM/400 data telephone number or set up PM/400 to use Advantis (or IGN - IBM Global Network). In releases of PM/400 prior to Version 3 Release 6, the CHGDTAARA command on QMPGLIB/Q1PGTELE (for changing the PM/400 data telephone number) or QMPGLIB/Q1PCTL (for changing the RMTLOCNAM, RMTNETID, and LCLLOCNAM) must be used.

10.2.6 Display PM/400 Status

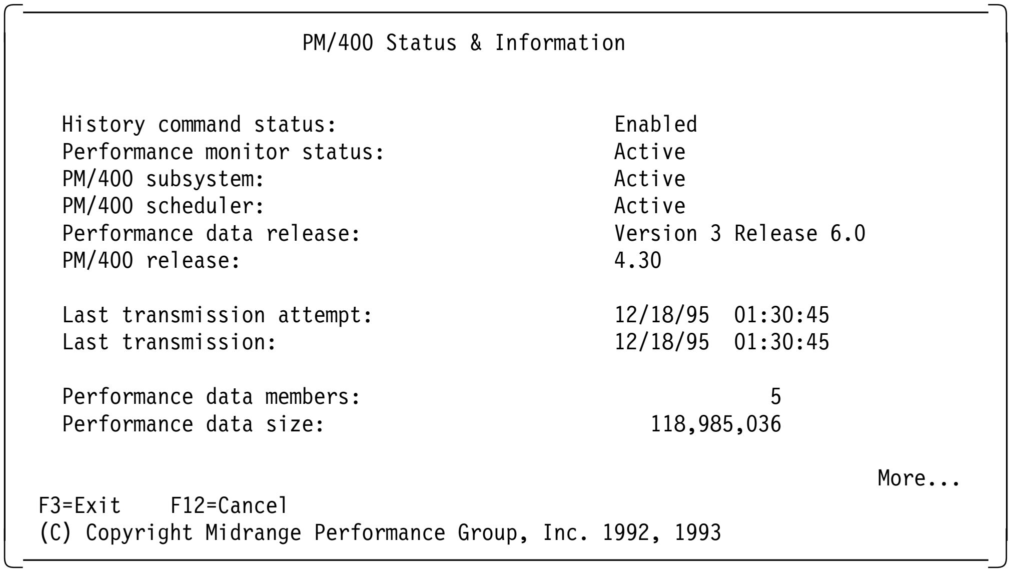
To reach the PM/400 Status and Information display, take Option 9 in Figure 35 on page 155.



```
PM/400                     PM/400 Menu

Select one of the following:
     1. Work with Historical System Status
     2. Work with Historical Active Jobs
     3. Work with Historical Disk Status
     4. Work with Contact/Enrollment Information
     5. Work with Automatically Scheduled Jobs
     6. Work with PM/400 Customization
     7. Work with Top Ten Omissions
     8. Work with Remote AS/400′s
     9. Display PM/400 Status


Selection or command
===> 9_______________________________

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=User support      F16=System main menu
```

Figure 35. PM/400 Menu - Display 3

The new ″Display PM/400 Status″ menu option not only allows you to determine the status of many of the PM/400 functions, but it also allows you to review a three-month history of your system′s trend information.



```
                      PM/400 Status & Information


  History command status:                      Enabled
  Performance monitor status:                  Active
  PM/400 subsystem:                            Active
  PM/400 scheduler:                            Active
  Performance data release:                    Version 3 Release 6.0
  PM/400 release:                              4.30

  Last transmission attempt:                   12/18/95  01:30:45
  Last transmission:                           12/18/95  01:30:45

  Performance data members:                                5
  Performance data size:                          118,985,036

                                                                More...
F3=Exit    F12=Cancel
(C) Copyright Midrange Performance Group, Inc. 1992, 1993
```

Figure 36. PM/400 Status & Information - Display 4

From the Command Entry display, you can start the Display PM/400 Status function by typing the following command at the command line and pressing the Enter key:

```
GO QMPGLIB/PM400
```

Type **9** on the command line of the PM/400 Menu display and press the Enter key (see Display 3). The PM/400 Status & Information display is shown (see Display 4).

- **History Command Status:**

  This describes the status of the PM/400 history commands, which is determined by the level of PM/400 support that the customer is receiving. The PM/400 history commands are only activated if the customer is subscribing to the PM/400 fee service offering and actively transmitting collected data electronically to IBM (or if it is within 30 days after the first CFGPM400 command is run for the ″PM/400-Subset Service Exclusive″ customers).

- **Performance Monitor Status:**

  This indicates the current status of the OS/400 Performance Monitor (instead of having to use WRKACTJOB to check on the status of the QPFRMON job).

- **PM/400 Subsystem:**

  This indicates the current status of the PM/400 subsystem instead of having to use WRKACTJOB or WRKSYSSTS to check on the status of the QMPGSCH or Q1PGSCH subsystem (depending on the release level of the PM/400 code).

- **PM/400 Scheduler:**

  This indicates the current status of the PM/400 scheduler instead of having to use WRKACTJOB or WRKSBSJOB to check on the status of the QMPGSCH or Q1PGSCH job.

- **Performance Data Release:**

  This indicates the version, release, and modification of the performance monitor data. It should correspond to the version, release, and modification level of the OS/400.

- **PM/400 Release:**

  This is the PM/400 product release number.

- **Last transmission attempt:**

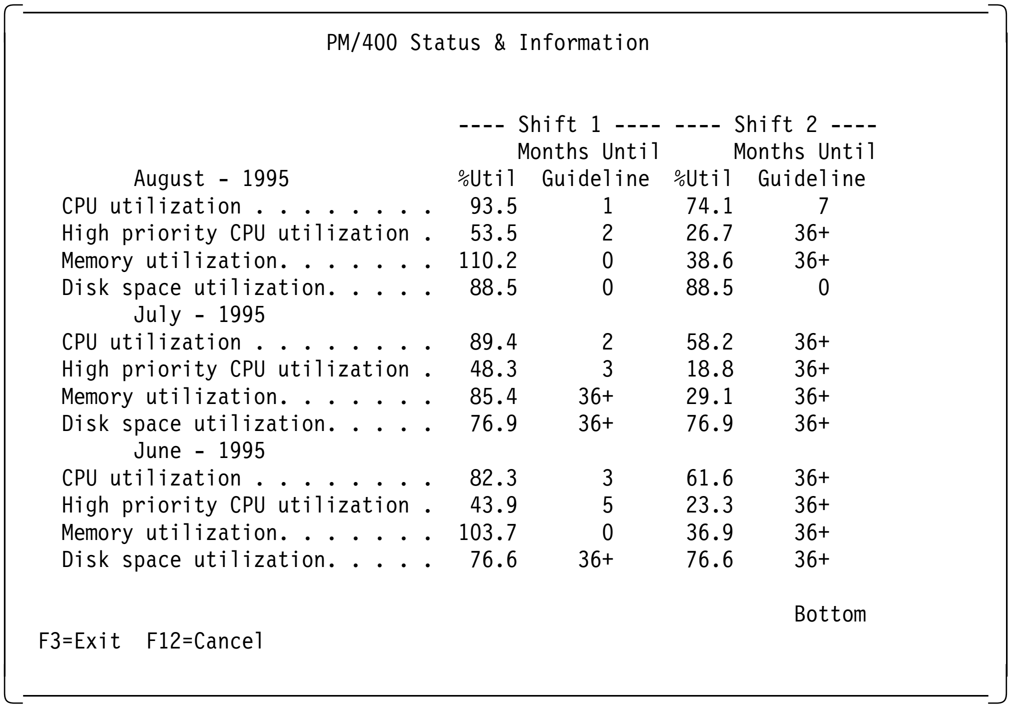
  This is the date and time of the last attempt to transmit the PM/400 data. By comparing this date and time with that of the last transmission, the customer can tell if the last transmission attempted was successful or not. This was not possible on the previous releases of PM/400.

- **Last transmission:**

  This is the date and time of the last completed transmission of the PM/400 data.

The PM/400 Status & Information display shows a three-month history of the system′s trend information. This is available for all PM/400 customers (both the PM/400 service offering customer and the PM/400 Subset customers).

As long as the customer has been collecting and transmitting data electronically to IBM, the IBM central site processes the transmitted data at the beginning of each month. This processed data is transmitted back to the customer′s system during the next transmission initiated from the customer′s system. Information on resource utilization (up to the last three months) is shown on the PM/400 Status & Information display.



```
                    PM/400 Status & Information


                               ---- Shift 1 ---- ---- Shift 2 ----
                                    Months Until      Months Until
        August - 1995          %Util  Guideline  %Util  Guideline
  CPU utilization . . . . . . . .  93.5       1      74.1       7
  High priority CPU utilization .  53.5       2      26.7     36+
  Memory utilization. . . . . . . 110.2       0      38.6     36+
  Disk space utilization. . . . .  88.5       0      88.5       0
        July - 1995
  CPU utilization . . . . . . . .  89.4       2      58.2     36+
  High priority CPU utilization .  48.3       3      18.8     36+
  Memory utilization. . . . . . .  85.4     36+      29.1     36+
  Disk space utilization. . . . .  76.9     36+      76.9     36+
        June - 1995
  CPU utilization . . . . . . . .  82.3       3      61.6     36+
  High priority CPU utilization .  43.9       5      23.3     36+
  Memory utilization. . . . . . . 103.7       0      36.9     36+
  Disk space utilization. . . . .  76.6     36+      76.6     36+

                                                          Bottom
F3=Exit  F12=Cancel
```

Figure 37. PM/400 Status & Information - Display 5

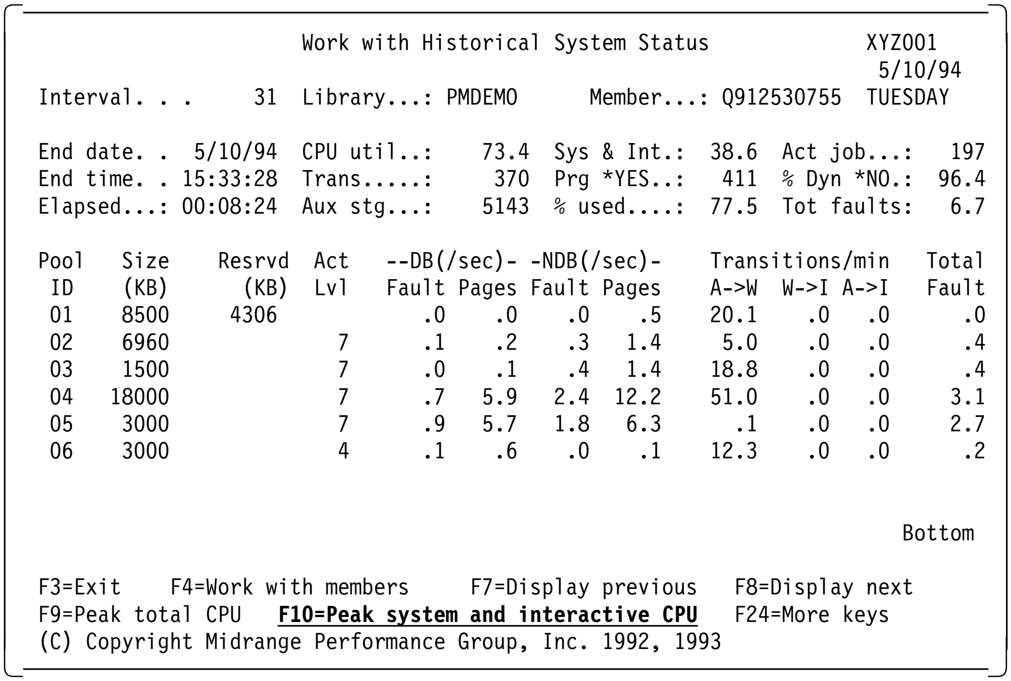
With this new function, those PM/400 customers can review CPU, memory, and DASD utilization and trend information online without having to wait for their PM/400 report to arrive. This may take a nominal time, depending on the service level to which they have subscribed.

10.2.7 Work with Historical Performance Data Functions of PM/400

These functions use performance data to create displays similar to the OS/400 Work System Status display, Work Active Jobs display, and Work Disk Status display. These displays quickly provide information useful in understanding historical events and conditions on the system and determining the appropriate corrective action. Using the enhanced function keys and search capabilities of the Work with Historical Performance Data functions, you can quickly locate the following information:

- A period of high total CPU utilization
- A period of high system and interactive CPU utilization
- Jobs using high CPU or high I/O
- A particular user or group of users consuming huge amount of resources

An example of the Work with Historical Performance Data functions is shown in Figure 38 on page 158.



```
                         Work with Historical System Status           XYZ001
                                                                      5/10/94
Interval. . .     31  Library...: PMDEMO      Member...: Q912530755  TUESDAY

End date. .  5/10/94  CPU util..:    73.4  Sys & Int.:  38.6  Act job...:   197
End time. . 15:33:28  Trans.....:     370  Prg *YES..:   411  % Dyn *NO.:  96.4
Elapsed...: 00:08:24  Aux stg...:    5143  % used....:  77.5  Tot faults:   6.7

Pool   Size    Resrvd  Act   --DB(/sec)- -NDB(/sec)-    Transitions/min   Total
 ID    (KB)      (KB)  Lvl   Fault Pages Fault Pages    A->W  W->I A->I   Fault
 01    8500      4306           .0    .0    .0    .5    20.1    .0   .0      .0
 02    6960              7      .1    .2    .3   1.4     5.0    .0   .0      .4
 03    1500              7      .0    .1    .4   1.4    18.8    .0   .0      .4
 04   18000              7      .7   5.9   2.4  12.2    51.0    .0   .0     3.1
 05    3000              7      .9   5.7   1.8   6.3      .1    .0   .0     2.7
 06    3000              4      .1    .6    .0    .1    12.3    .0   .0      .2



                                                                         Bottom

F3=Exit    F4=Work with members     F7=Display previous   F8=Display next
F9=Peak total CPU   F10=Peak system and interactive CPU   F24=More keys
(C) Copyright Midrange Performance Group, Inc. 1992, 1993
```

Figure 38. Sample Work with Historical System Status Display

10.2.8 Monthly PM/400 Graphs

These graphs and reports give management a clear understanding of the current performance of their AS/400 system. For example, the PM/400 graphs (a sample of which is shown in Figure 39 on page 159) show a 24-hour window on the four time graphs. With the 24-hour review, you can do the following:

- Understand the off-shift resource utilizations.
- Determine if overnight operations are starting to run into first-shift operations.
- Determine if the off-shift operations are resource constrained.

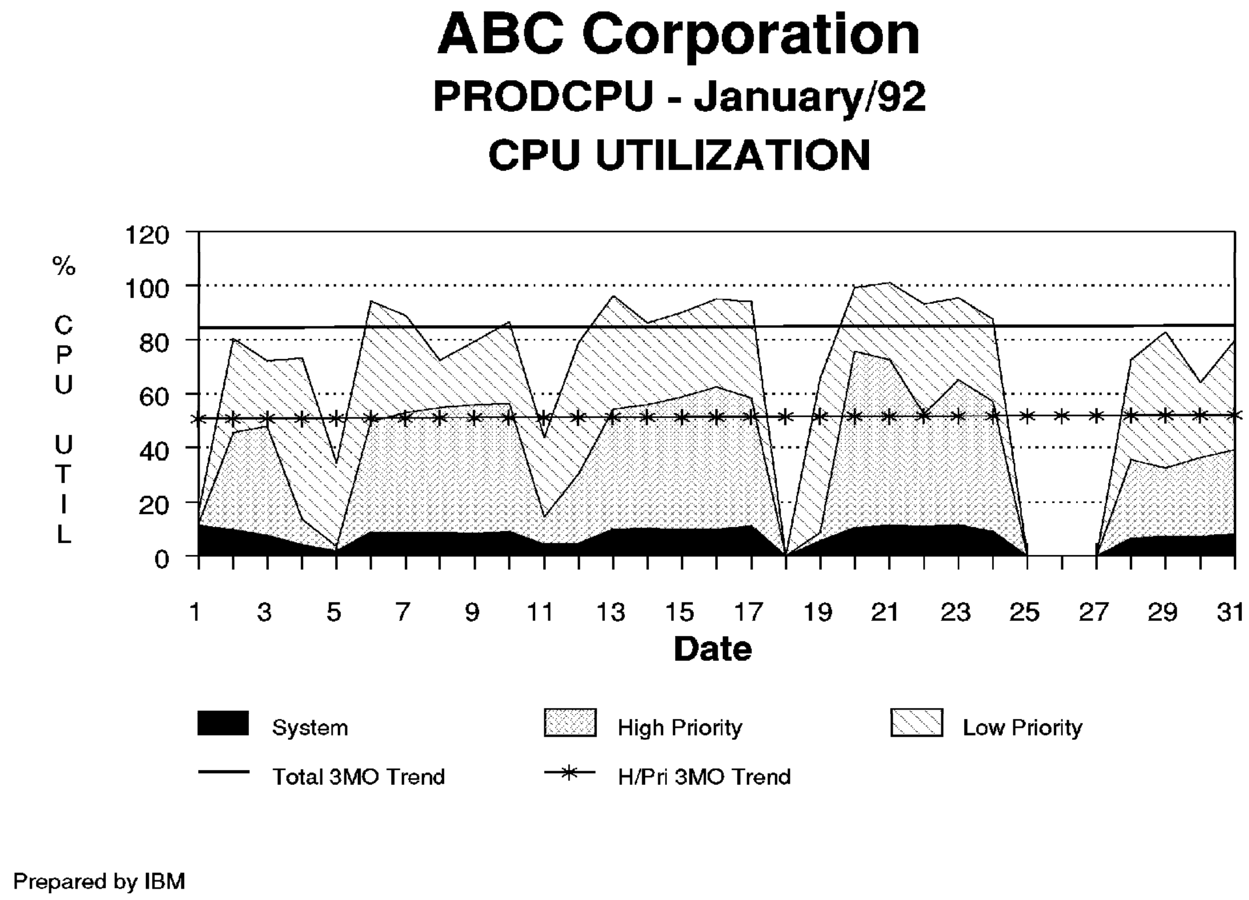
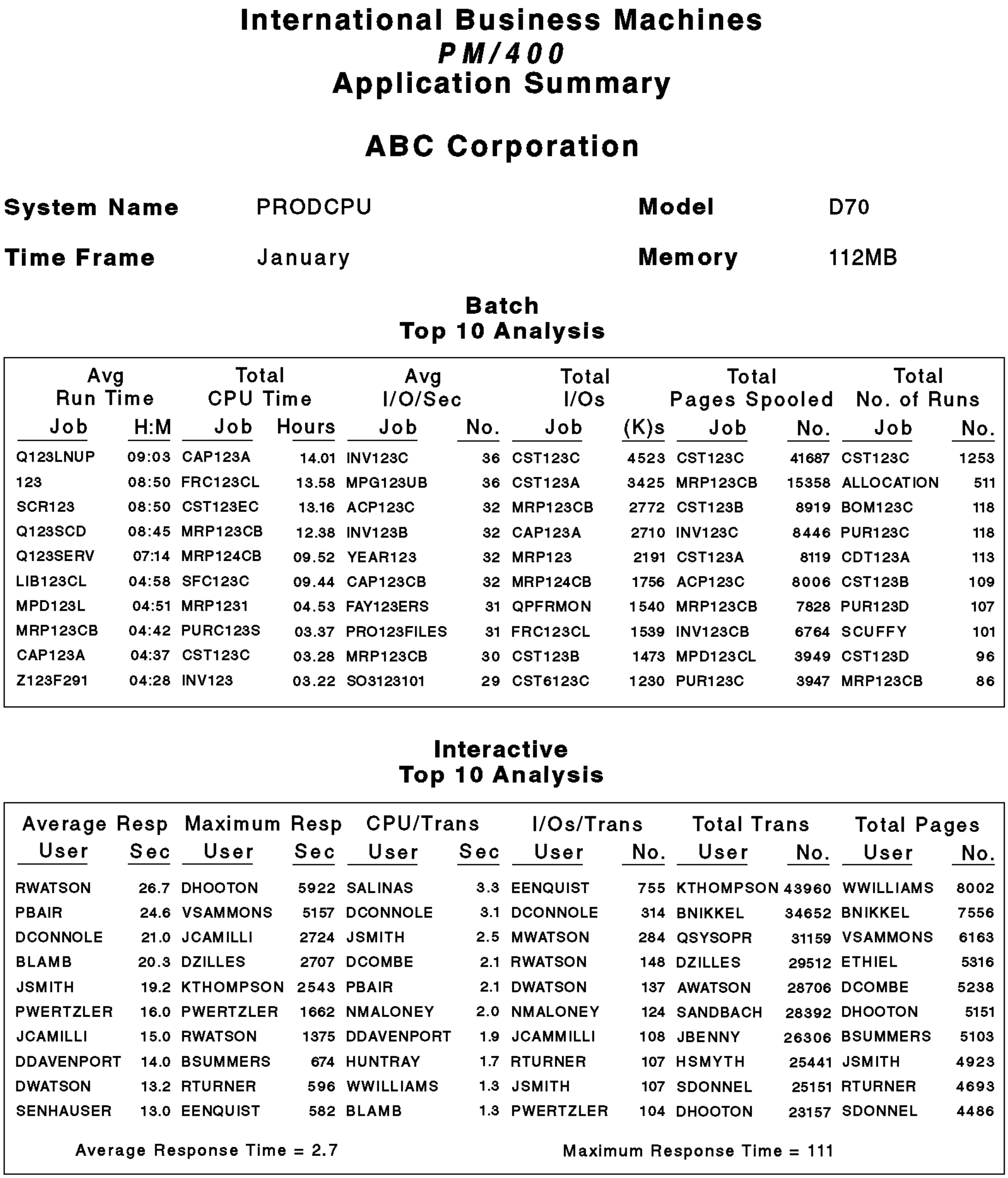


Figure 39. Sample PM/400 CPU Utilization Chart

PM/400 Application Summary Report: This report is designed to give the data processing manager a list of jobs and users that are using the most resources. This is accomplished by performing a Top 10 Analysis in six categories on batch jobs and interactive users. By understanding which jobs and users are using the most resources, often limited staff resources can be focused on the application area with the biggest potential payback.



International Business Machines
PM/400
Application Summary

ABC Corporation

System Name	PRODCPU	**Model**	D70
Time Frame	January	**Memory**	112MB

Batch
Top 10 Analysis

Avg Run Time		Total CPU Time		Avg I/O/Sec		Total I/Os		Total Pages Spooled		Total No. of Runs	
Job	H:M	Job	Hours	Job	No.	Job	(K)s	Job	No.	Job	No.
Q123LNUP	09:03	CAP123A	14.01	INV123C	36	CST123C	4523	CST123C	41687	CST123C	1253
123	08:50	FRC123CL	13.58	MPG123UB	36	CST123A	3425	MRP123CB	15358	ALLOCATION	511
SCR123	08:50	CST123EC	13.16	ACP123C	32	MRP123CB	2772	CST123B	8919	BOM123C	118
Q123SCD	08:45	MRP123CB	12.38	INV123B	32	CAP123A	2710	INV123C	8446	PUR123C	118
Q123SERV	07:14	MRP124CB	09.52	YEAR123	32	MRP123	2191	CST123A	8119	CDT123A	113
LIB123CL	04:58	SFC123C	09.44	CAP123CB	32	MRP124CB	1756	ACP123C	8006	CST123B	109
MPD123L	04:51	MRP1231	04.53	FAY123ERS	31	QPFRMON	1540	MRP123CB	7828	PUR123D	107
MRP123CB	04:42	PURC123S	03.37	PRO123FILES	31	FRC123CL	1539	INV123CB	6764	SCUFFY	101
CAP123A	04:37	CST123C	03.28	MRP123CB	30	CST123B	1473	MPD123CL	3949	CST123D	96
Z123F291	04:28	INV123	03.22	SO3123101	29	CST6123C	1230	PUR123C	3947	MRP123CB	86

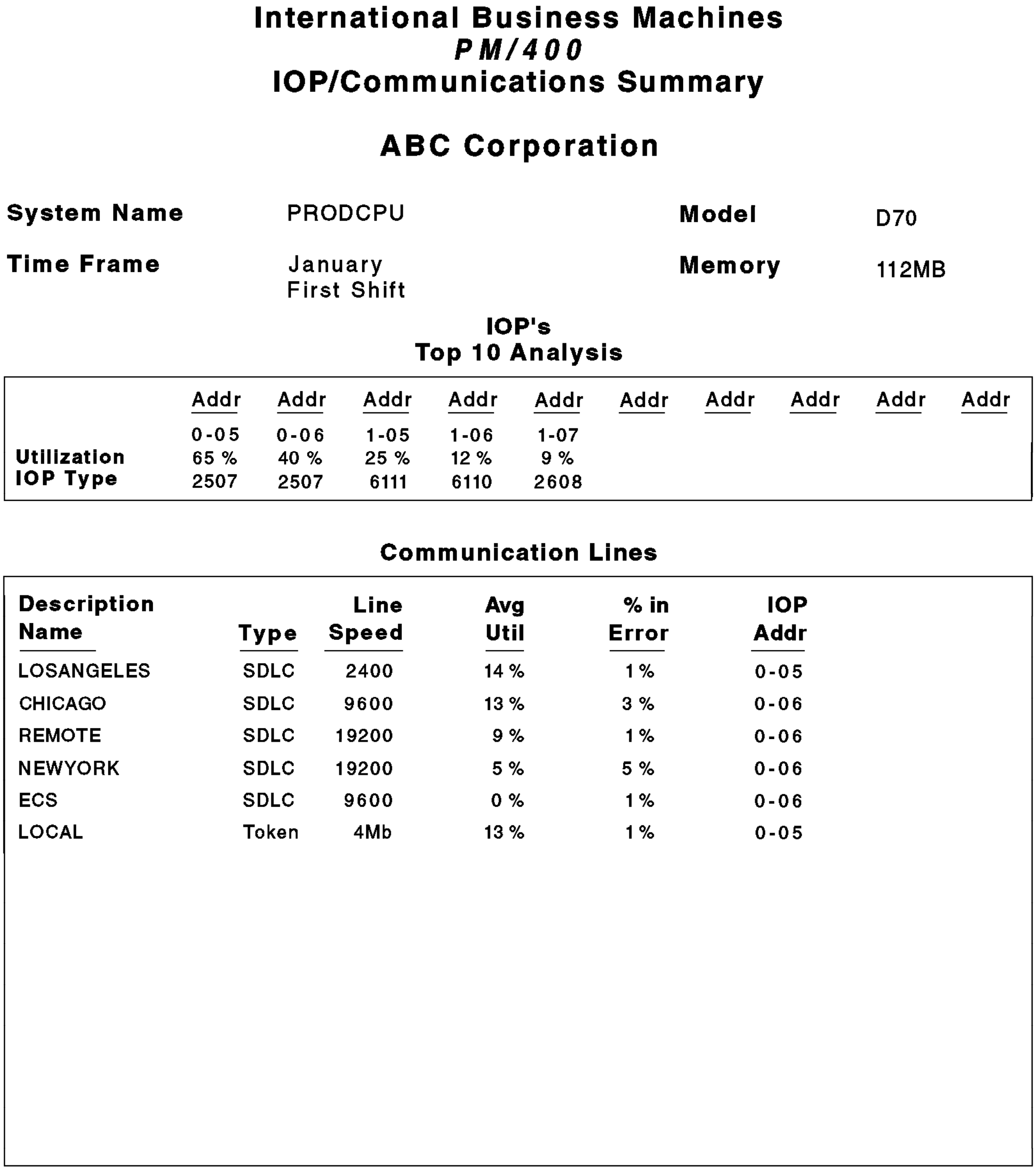
Interactive
Top 10 Analysis

Average Resp		Maximum Resp		CPU/Trans		I/Os/Trans		Total Trans		Total Pages	
User	Sec	User	Sec	User	Sec	User	No.	User	No.	User	No.
RWATSON	26.7	DHOOTON	5922	SALINAS	3.3	EENQUIST	755	KTHOMPSON	43960	WWILLIAMS	8002
PBAIR	24.6	VSAMMONS	5157	DCONNOLE	3.1	DCONNOLE	314	BNIKKEL	34652	BNIKKEL	7556
DCONNOLE	21.0	JCAMILLI	2724	JSMITH	2.5	MWATSON	284	QSYSOPR	31159	VSAMMONS	6163
BLAMB	20.3	DZILLES	2707	DCOMBE	2.1	RWATSON	148	DZILLES	29512	ETHIEL	5316
JSMITH	19.2	KTHOMPSON	2543	PBAIR	2.1	DWATSON	137	AWATSON	28706	DCOMBE	5238
PWERTZLER	16.0	PWERTZLER	1662	NMALONEY	2.0	NMALONEY	124	SANDBACH	28392	DHOOTON	5151
JCAMILLI	15.0	RWATSON	1375	DDAVENPORT	1.9	JCAMMILLI	108	JBENNY	26306	BSUMMERS	5103
DDAVENPORT	14.0	BSUMMERS	674	HUNTRAY	1.7	RTURNER	107	HSMYTH	25441	JSMITH	4923
DWATSON	13.2	RTURNER	596	WWILLIAMS	1.3	JSMITH	107	SDONNEL	25151	RTURNER	4693
SENHAUSER	13.0	EENQUIST	582	BLAMB	1.3	PWERTZLER	104	DHOOTON	23157	SDONNEL	4486

Average Response Time = 2.7 Maximum Response Time = 111

Figure 40. Sample PM/400 Application Summary Report

PM/400 IOP/Communications Summary Report: This report, shown in Figure 41 on page 161, lists up to 10 communications lines by percentage of utilization. Additionally, the percentage of data that was either transmitted or received in error is listed. This report alerts you to over-utilization or error conditions that cause performance problems.



International Business Machines
PM/400
IOP/Communications Summary

ABC Corporation

System Name	PRODCPU	**Model**	D70
Time Frame	January First Shift	**Memory**	112MB

IOP's
Top 10 Analysis

	Addr	Addr	Addr	Addr	Addr	Addr	Addr	Addr	Addr	Addr
	0-05	0-06	1-05	1-06	1-07					
Utilization	65 %	40 %	25 %	12 %	9 %					
IOP Type	2507	2507	6111	6110	2608					

Communication Lines

Description Name	Type	Line Speed	Avg Util	% in Error	IOP Addr
LOSANGELES	SDLC	2400	14 %	1 %	0-05
CHICAGO	SDLC	9600	13 %	3 %	0-06
REMOTE	SDLC	19200	9 %	1 %	0-06
NEWYORK	SDLC	19200	5 %	5 %	0-06
ECS	SDLC	9600	0 %	1 %	0-06
LOCAL	Token	4Mb	13 %	1 %	0-05

Figure 41. Sample PM/400 IOP/Communications Summary Report

10.2.9 PM/400 Tips and Techniques

The following list contains some general tips:

- PM/400 (5716-PM1) must always be installed using the ***QSECOFR*** profile or user authority.
- The use of the objects in PM/400 (for example, Q1PDR that starts QPFRMON) is dependant on QPGMR having the correct authority. Many customers only allow QPGMR to have only *JOBCTL and *SPLCTL; this may not be sufficient. PM/400 requires that the customer grant object authority (GRTOBJAUT) for QPGMR to the two main libraries (QMPGLIB and QMPGDATA). Also QPGMR needs authority to the start performance monitor (STRPFRMON) command. The IBM PM/400 team is releasing a PTF that creates a special user profile to run the PM/400 objects, but QPGMR still requires authorized access to certain system and PM/400 objects. This PTF is currently being tested.
- If a customer transmits to the PM/400 factory site but there is no data in the member transmitted, check to see if the Performance Monitor is running under the subsystem Q1PGSCH. If it is running successfully, check to see that the data reduction process is working correctly by looking at the history logs, job logs, or by looking at QSYSOPR message queue to see if it ran successfully. Another method is to change the *next run date and time* parameter in the menu to run at that moment so the messages can be checked. To do this, go to the PM400 menu (GO QMPGLIB/PM400) and take option 5 (WORK WITH AUTOMATICALLY SCHEDULED JOBS). Select option 2 to change job ***Q1PDR*** to run at a convenient time to view the results. Using WRKSBSJOB Q1PGSCH, watch the jobs running under Q1PGSCH. The standard jobs running should be QPRMON and Q1PSCH. The Q1PDR job starts to run after the time has elapsed. If it ends with a *JOBLOG, review the job log and the QSYSOPR message queue for the reason. Common causes for incorrect operation are:
  - QPGMR does not have authority to the objects.
  - The performance data has not been converted from a previous release. This can be remedied using the CVTPFRDTA command,

CPF1136 Error Messages in the JOBLOG: The reason is caused by duplicate entries in the job descriptions for Q1PSCH1, Q1PSCH2, and Q1PSCH3.

To eliminate the problem, remove the duplicate entries, usually ***QGPL or QTEMP***.

Communication Problems: First check the job logs for ***Q1PCM1*** for any messages of VARY ON/OFF failures.

Check the Q1PDEV, Q1PCTL, and Q1PLIN configuration objects for incorrect addresses. The Q1PDEV parameter is LCLLOCNAME = CLIENTxx; the Q1PCTL and Q1PLIN parameters are STNADDR = xx (from the Q1PDEV parameter). This can be caused by running the CFGPM400 program when creating a new device configuration object and not deleting the objects that existed.

Install Problems: The product *must* be installed under ″QSECOFR″ authority.

If the customer has renamed the ***QESLIN (ECS line)***, this causes the INSTALL operation to have problems.

Either have the customer create a new line, controller, and device called QECLIN, QESCTL, and QESDEV, and run the install again, or rename them to the original names (QESxxx) and run the install process again. When the install has completed successfully, the configuration objects can be renamed back again.

If ***CFGPM400*** failed with MSGCPD0020/14/13/78 message information `Character ′x′ not valid following string xx-xx. Character ′ ′ not valid following string ′xx.` or similar information, check the contact information (WRKCNTINF). Change or delete the information in OPTION SERVICE INFO.

10.3 Performance Investigator/400 (PI/400)

The Performance Investigator/400 (PI/400) was originally introduced as a PRPQ (5799-PRG) with AS/400 V2R2 and ran on V2R3 under Windows and retrieved data from an AS/400 system using APPC through PC Support/400.

This section describes the **new** Performance Investigator, which, while maintaining the capabilities of the old Performance Investigator, provides additional performance monitoring, improved graphical interaction, and user exit program functions.

10.3.1 Performance Investigator/400 Overview

Typical resources that operations want to monitor include the utilization of CPU, disk hardware, communications lines, and I/O Processors or a specific ″measurement item″ important to understanding performance on a system, such as interactive job response time or logical I/O counts.

Performance Investigator/400 now supports monitoring of any number of AS/400 systems, though you are still limited by the display technology to actually view ″any number of AS/400 systems″ on a display at the same time.

PI/400 is aimed at customers who have sufficient staffing to visually monitor the display and develop intelligent user-exit programs that do some kind of automation. Multiple AS/400 host systems can be monitored by a single PI/400 client workstation and up to two PI/400 client workstations can monitor the same host AS/400 system.

IBM Performance Investigator/400 combines AS/400 real-time performance data collection with a PC-based graphical view of that data. Data is collected from the AS/400 system in samples as often as every 15 seconds (but typically five-minute intervals) and sent to the PC for threshold analysis and display. A threshold is defined in PI/400 as a performance value where the CPU utilization is equal to or exceeds a specific value for a nominated length of time

This tool provides the user with immediate feedback on the performance of their system (or systems), highlighting problem areas where user definable thresholds are exceeded while using minimal host CPU resource.

A threshold can also call a program on either the AS/400 system or the PI/400 client workstation.

Figure 42 on page 164 shows an example of PI/400 client workstation display.

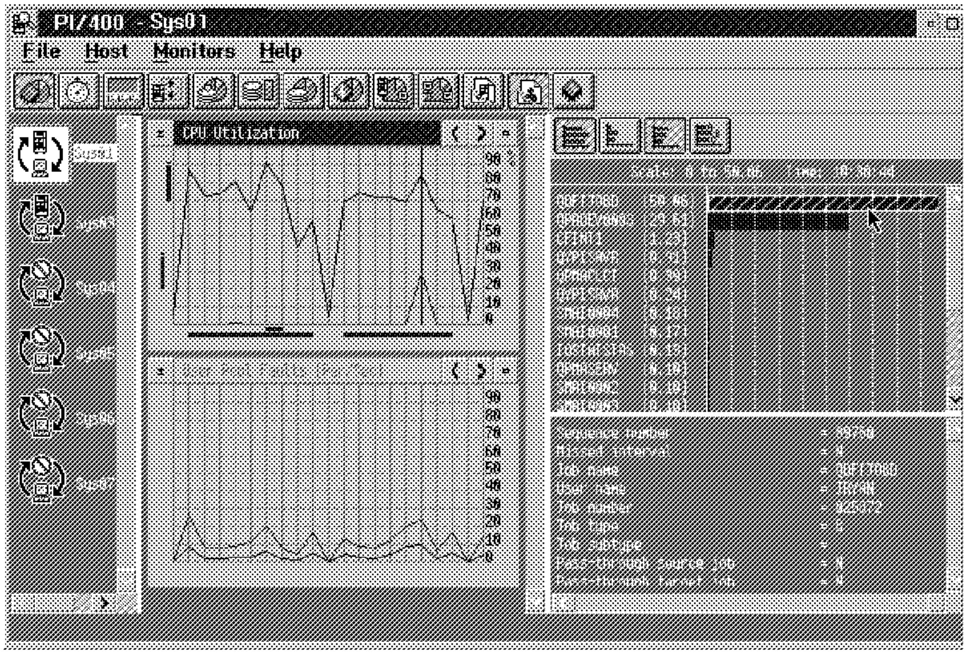


Figure 42. PI/400 Three-Level CPU Monitoring

In the sample windows shown in Figure 42, we see the CPU utilization for Sys01. Using the arrows at the top of the CPU Window, you can go back in time for several hours.

The window under CPU utilization shows User Pool Faults.

If you click on the icons on the action bar, you can open windows for each kind of performance metric you want PI/400 to monitor.

There are up to three levels of performance data possible. For example, for CPU utilization:

- Level 1 shows summary information for several time intervals.
- You can click on a specific time interval and get level 2 data, which shows CPU utilization for jobs sorted according to highest CPU utilization by a job. (This is shown in the upper window on the right.) For CPU utilization (Interactive response time, Transaction rate, and Batch logical I/O counts), the top 20 jobs are shown.
- You can get level 3 data for a job by clicking on a particular job. This gives you more information about the job (for example, job number, user name, job name, and so on).

This example shows all three levels of PI/400 graphical information for CPU utilization. The diagonal lines in the horizontal bar for job QDTFJOBD indicate which job was selected for level 3 (specific job) information.

10.3.2 PI/400 Hardware and Software Requirements

In order to use PI/400, the following hardware and software is required:

- OS/400 must be at V3R1, V3R2, V3R6, or V3R7
- V3R1: 5799-QGY + PTFs (PRPQ)
- V3R2: 5799-QGY (PRPQ)
- V3R6 + V3R7: 5799-FRD (PRPQ)
- TCP/IP Function of OS/400 must be active and FTP must be enabled.
- For V3R1, 5799-XDB (CPA Extensions for OS/400) is required.
- OS/2 workstations require OS/2 Warp Version 3.0 or later.
- When monitoring multiple systems, additional PC capacity is required. A PC larger than a 486 33Mhz processor and 16MB of memory is recommended.
- OS/2 TCP/IP Version 2.0 or later
- PI/400 host code must be installed on each host AS/400 system you want to monitor.

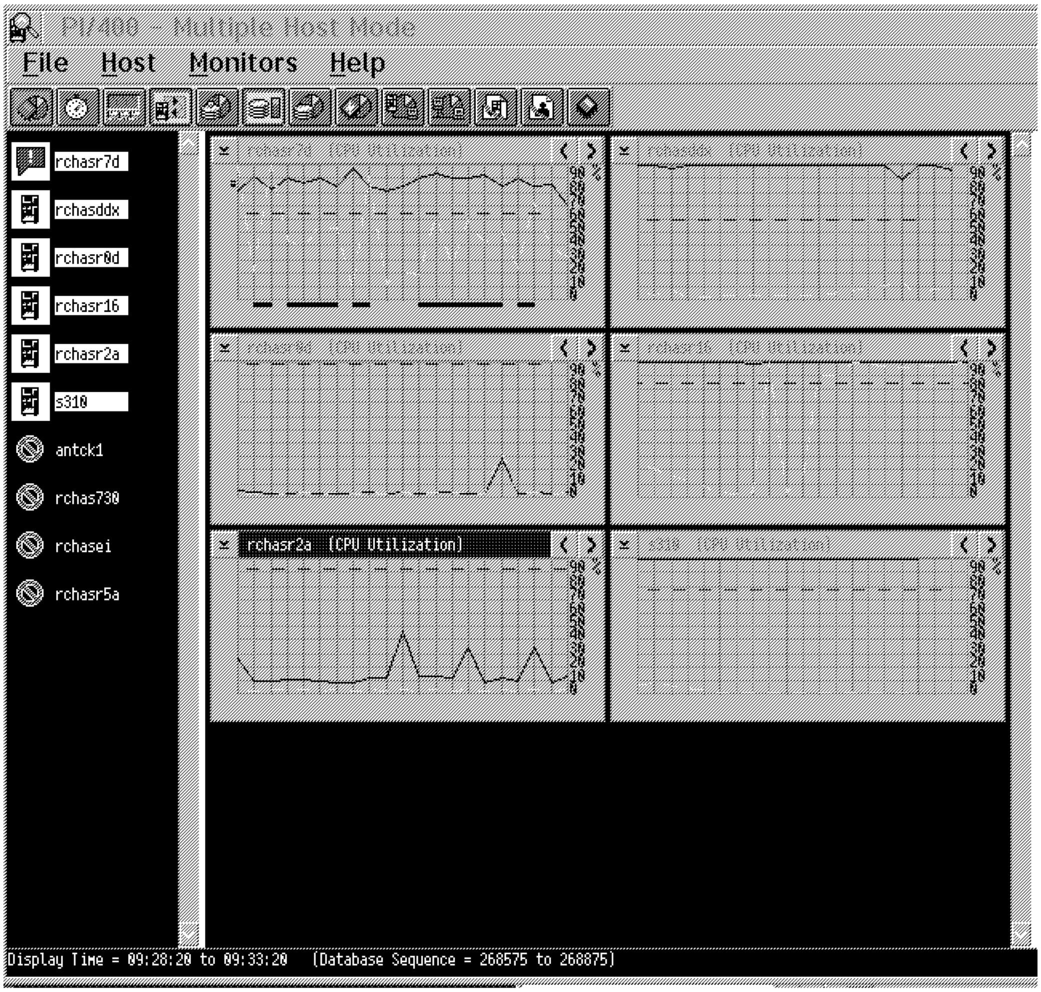
Note: PI/400 does not run on a system configured for C2 security.

For details on PI/400 installation, see *IBM Performance Investigator for OS/400*, SC41-0624.

10.3.3 Multi-Host Mode

In multi-host mode, you can display graphs for multiple AS/400 host systems side-by-side on one workstation. Each Level 1 graph represents one host, and the graph shows the monitor lines for each type of data being monitored for that host.

Each of the monitors you select for that host is represented by a line on the graph for that host. You can set the colors for the lines on the graph for each host. If you forget what a color represents, you can display a legend that shows what all the colors mean.


PI/400 - Multiple Host Mode
File
Host
Monitors
Help
rchasr7d
rchasddx
rchasr0d
rchasr16
rchasr2a
s310
antck1
rchas730
rchasei
rchasr5a
rchasr7d (CPU Utilization)
rchasddx (CPU Utilization)
rchasr0d (CPU Utilization)
rchasr16 (CPU Utilization)
rchasr2a (CPU Utilization)
s310 (CPU Utilization)
90
80
70
60
50
40
30
20
10
0
%

Display Time = 09:28:20 to 09:33:20 (Database Sequence = 268575 to 268875)

Figure 43. Sample PI/400 Multi-Host Window

In Figure 43, we have sized the windows so that performance data for up to eight systems can appear on the window at the same time. In this case, six AS/400 systems are selected and are shown at the monitor. There are another four systems that you can select. They were not currently online when this sample was taken.

10.3.4 Thresholds

You can use PI/400 to visually monitor performance (CPU utilization, interactive response time, and so on). You can create a threshold definition that allows calling a program or operator notification when a trigger specified for the threshold occurs.

A system administrator can determine the action to be taken without operator intervention. A program can be written that does things such as sending an SNA alert or SNMP trap, dialing a pager, starting or holding a job, and so on.

Each threshold has:

- Element 1 - Threshold attribute:
  - *AVG (default):

Thresholds are compared against an average value for the collected data (performance metric).

  - *MAX:

    Thresholds are compared against a maximum value for the performance metric.

  - *INTERACT:

    Thresholds are compared against an interactive value for the performance metric that corresponds to an attribute of the metric. For example, when specifying TYPE(*CPU), PI/400 compares a threshold against the average CPU utilization for **only interactive jobs.**

- Element 2 - Trigger value:
  - *NONE (default):

    No value is used to trigger a threshold. If *NONE is specified, threshold monitoring is not performed.

  - Trigger value (1-99999999).

- Element 3 - Trigger activate duration:
  - 1 (default):

    The number of consecutive intervals (INTERVAL keyword on STRPICOL) that the trigger value must be maintained in order for the trigger action to be performed.

  - Trigger-duration (1-99999999 consecutive intervals)

- Element 4 - Trigger action:
  - *NONE (default):

    Do not perform any action when a threshold value is triggered.

  - Trigger-action:

    Specify a valid action (command or CALL program) that is performed when a combination of trigger value and trigger activate duration (consecutive intervals) is met.

When there is no programmed automatic action defined and a threshold is triggered, PI/400 alerts the operator of the condition by a flashing PI/400 window. It is up to the operator to determine what action should be taken. If you can identify operator actions that can be performed through user programs, you can automate these actions.

With the old PI/400, there was no interface to allow calling a user program. Also, the new PI/400 is built on OS/2 because of the stability of OS/2 in managing multiple tasks at the same time.

While OMEGAMON/400 with its automated facilities enables calling a user program, it also provides monitoring of much more than just performance-oriented conditions and some ″predefined″ actions for holding a job or starting a job, and so on.

However, the CPU consumption of OMEGAMON using Automated Facilities for performance information is significantly more than using the PI/400 but PI/400 displays the performance data graphically in real time. With OmegaView, you get an ICON ″colored″ that indicates a threshold has been reached, but no display of performance information details as is available with PI/400.

When a threshold has been reached, not only can an OS/400 program be called but you can also call a program on the PI/400 workstation.

- Element 5 - Reset value:

  This is when a trigger threshold has been reached and a compare value to ″reset″ the threshold is reached afterwards. For example, after the CPU utilization reaches 90%, (a trigger threshold) program xxxx is called. After this, CPU utilization reaches 70% and we need to ″undo″ what that program xxxx did. So 70% utilization is used as a threshold ″reset″ and program yyyy is called.

  – *TRGVAL (default):

    The reset value is the same as the trigger value.

  – *NONE:

    No value is used to reset a threshold.

  – Reset-value:

    Specify a value (1-99999999) that causes the trigger threshold to be reset.

- Element 6 - Reset activate duration:

  – *TRGDUR (default):

    The reset duration is the same as the trigger duration.

  – Reset-duration:

    Specify a value (1-99999999) to set the number of consecutive intervals required at or above the reset value before the reset action occurs.

- Element 7 - Reset action:

  – *NONE (default):

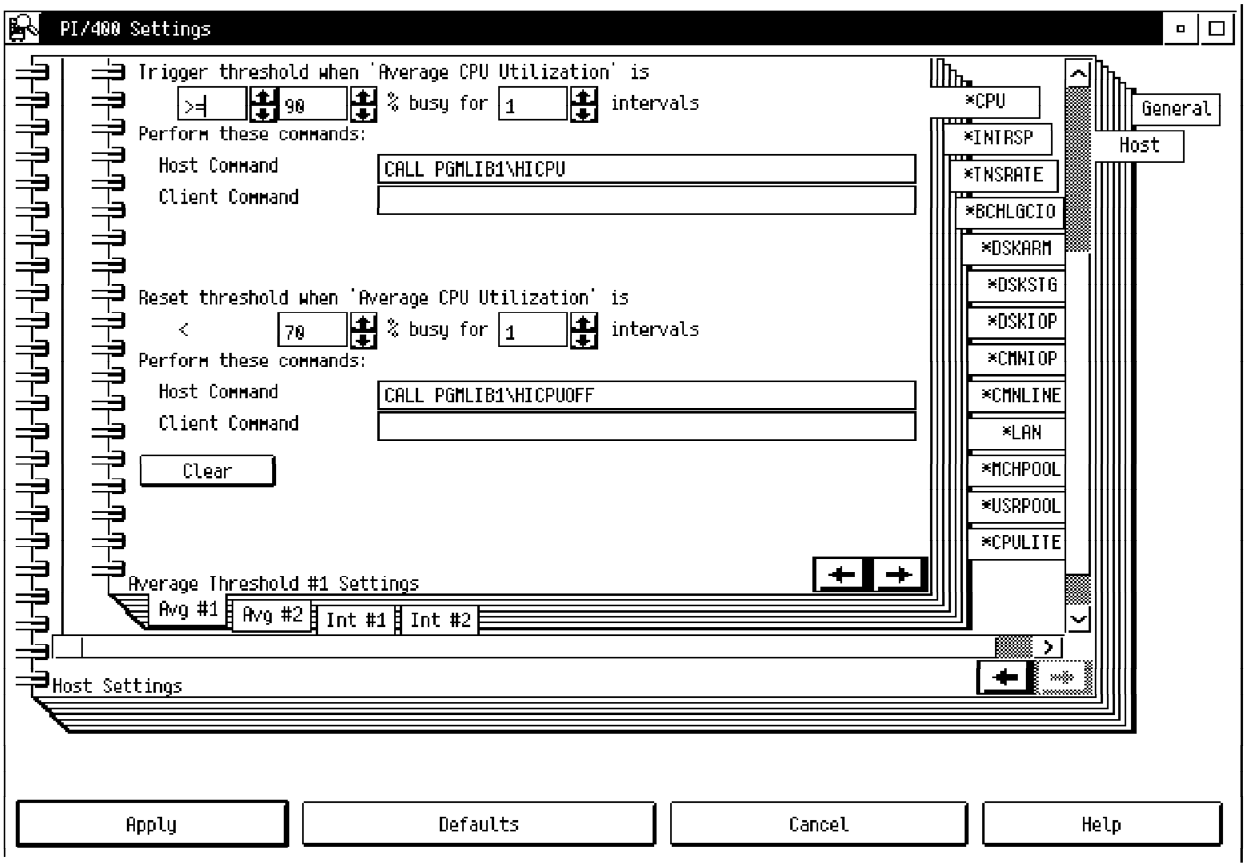
    Do not perform any action when a threshold value is reset.

  – Reset-action:

    Specify a valid action (command or CALL program) that is performed when a combination of the reset value and reset activate duration (consecutive intervals) is met.

Figure 44 on page 169 shows an example of how to set up a trigger and a reset condition under the following circumstances:

- When CPU utilization reaches 90%, and remains at 90% or more than one minute, a procedure to block any new jobs from starting is automatically initiated.
- When CPU utilization drops to 70%, the blocking of new jobs is suspended.



Trigger threshold when 'Average CPU Utilization' is
>= 90 % busy for 1 intervals
Perform these commands:
Host Command CALL PGMLIB1\HICPU
Client Command

Reset threshold when 'Average CPU Utilization' is
< 70 % busy for 1 intervals
Perform these commands:
Host Command CALL PGMLIB1\HICPUOFF
Client Command

Figure 44. Sample Configuration Window for PI/400 Triggers

This is an example of calling different programs when a CPU utilization threshold is reached over a specified time interval. Either an OS/400 command or program may be called or a client workstation program may be called. In any case, the customer or a service provider has to write the ″automated program″.

10.3.5 Types of Performance Data

For most types of performance data, you can collect two measurements so two lines are displayed on the graph in the PI/400 window. The measurements are labeled primary and secondary in the following table.

For example: Two lines are shown for CPU Utilization. The first one, called ″Primary″, shows the average CPU Utilization of the entire system. The second one, called ″Secondary″, shows the CPU Utilization by interactive Jobs.

Table 19 shows the types of performance data that you can monitor with PI/400.

Table 19 (Page 1 of 2). Performance Values That Can Be Monitored

Performance Data	**Primary**	**Secondary**
CPU Utilization *	Average	Interactive
Interactive response time *	Average	Maximum (job)
Transaction rate	Average	Interactive
Batch logical I/O	Average	Maximum
Disk arm Utilization *	Average	Maximum (arm)

Table 19 (Page 2 of 2). Performance Values That Can Be Monitored

Performance Data	Primary	Secondary
Disk Storage	Average (arm)	Maximum
Disk IOP Utilization *	Average	Maximum (IOP)
Communications IOP Utilization *	Average	Maximum (IOP)
Communications line Utilization *	Average	Maximum (line)
LAN Utilization *	Average	Maximum (line)
Machine pool faults *	Maximum	
User pool faults *	Average	Maximum (pool)
CPU Utilization (Lite)	Average	Interactive
Note: The performance values identified with a * were supported by the old Performance Investigator.		

Table 19 on page 169 shows the different types of data that can be collected. For most types of performance data, you can collect two measurements so two lines are displayed on the graph in the PI/400 window. The measurements are labeled primary and secondary in the table.

CPU Utilization This is the percentage of available processing unit time that is being consumed on the system. The primary monitor line shows the average CPU utilization for all jobs on the system. The secondary line shows the interactive CPU utilization.

Interactive Response Time This is the response time in seconds experienced by interactive jobs on the system. The primary line shows the average interactive response time and the secondary line shows the maximum interactive response time experienced by any specific interactive job.

Transaction Rate This is the number of transactions that are being completed on the system per second. The primary line shows the system wide transaction rate. The secondary line shows the transaction rate for interactive jobs only.

Batch Logical I/O This is the number of logical disk I/O operations performed per second by batch jobs on the system. The primary line shows the average batch logical I/O. The secondary line shows the maximum batch logical I/O.

Disk Arm Utilization The primary line shows the average utilization of all the disk arms on the system and the secondary line shows the maximum utilization of any disk arm on the system.

Disk Storage The primary line shows the average % full of all the disk arms on the system. The secondary line shows the maximum % full of any disk arm on the system.

Disk IOP Utilization The primary line shows the average utilization of all the disk IOPs on the system. The secondary line shows the maximum utilization of any individual disk IOP on the system.

Communication IOP Utilization This shows how busy the communication IOPS on the system are during the current interval. The primary line shows the average utilization of all the communication IOPs on the system. The secondary line shows the maximum utilization of any individual communication IOP on the system.

Communication Line Utilization The primary line shows the average utilization of all communication lines on the system. The secondary line shows the maximum utilization of any individual communication line on the system.

LAN Utilization The primary line shows the average utilization of all LAN lines on the system. The secondary line shows the maximum utilization of any individual LAN line on the system.

Machine Pool Faults This shows the level of faulting activity in the system′s machine pool. The primary line shows the number of faults per second in this pool. There is no secondary line.

User Pool Faults The primary line shows the average faults per second occurring in all of the user pools on the system. The secondary line shows the maximum faults per second occurring in any specific user pool on the system.

CPU Utilization Lite This is the percentage of available processing unit time that is being consumed on the system. It is similar to the CPU utilization, except that the level 2 and level 3 data is not available.

10.3.6 Summary of Benefits for PI/400

AS/400 administrators or system operators require tools to provide immediate and up-to-the-minute performance information about the systems they manage. For administrators or operators with multiple systems and networks, this information becomes even more important. As systems and networks become larger, the data about the critical system resources becomes much more difficult to gather, analyze, and react to.

Performance Investigator/400 combines AS/400 real-time performance data collection with a PC-based graphical view of data to meet this business need.

PI/400 can be used to ″review″ several periods of monitored data to determine if there was a momentary peak of resource utilization or poor performance, or if there is continued high utilization or poor performance.

PI/400 also enables user-programmed automation.

10.4 Queries/Programs

Performance Monitor is part of OS/400 and stores all of the information in externally described database files that are described in Appendix A, Performance Data in the *OS/400 Work Management Version 3*, SC41-4306.

If the Performance Tools program product is not available at your installation, or if you need specific performance information on a regular basis, you may consider writing your own programs or queries using the Performance Monitor database files.

For example, the following SQL statement displays all of the interactive jobs that are using more than 0.5 CPU seconds per transaction in each interval sorted by CPU per transaction.

```
SELECT DTETIM , JBNAME , JBUSER , JBNBR , (JBCPU/JBNTR/1000)
FROM QPFRDATA/QAPMJOBS
WHERE JBTYPE = ′I′ and JBNTR > 0 and (JBCPU/JBNTR/1000) > 0.5
ORDER BY 5 DESC
```

You have to be careful when defining your queries to consider the format of every field, unit of measure, meaning of each record, and exact meaning of every field involved. The difference between different fields is sometimes subtle and you may end up using the wrong field or a wrong function.

The main disadvantage of the ″rolling-your-own″ approach is that you need to maintain your application with every new release because the Performance Monitor database file structure may change from release to release.

Chapter 11. System Performance Tuning Tips

This chapter discusses the tuning concepts of the AS/400 system. The information in this chapter can be used to understand system work and general tuning techniques so that adequate resources are provided for jobs and system work. The basic idea of tuning is that you balance the usage of a system′s resources which are:

- CPU
- Main memory
- Disk storage
- Communication lines

This chapter discusses system jobs, system values, and system-supplied subsystem monitor performance-related parameters. Tips are provided for setting up the AS/400 system so that jobs can share the resources available with minimum conflicts. When you receive the AS/400 computer, the system values are set to default values and the subsystem descriptions (especially the pool definitions) may not be satisfactory for your processing needs.

This section also lists some changes to the list of IBM system jobs in V3R6 as compared to previous releases. *OS/400 Work Management Version 3*, SC41-4306, provides some of this information. Other sections of this redbook are referenced when additional information is provided on the system work.

Please note that aspects of tuning have also been discussed in Chapter 9, "Performance Problem Analysis" on page 127.

11.1 V3R6 System Jobs

Often, when you use some of the performance tools and utilities, you see system programs running. This section describes what many of these programs actually do and helps you understand OS/400.

11.1.1 System Start Control Program Functions Job (SCPF)

This system job is started during IPL and provides the environment and directs the functions necessary to start up OS/400. Most of its work is completed by the time workstation or non-interactive user jobs can be started.

SCPF remains active during normal system operation and most of the time consumes little CPU. Under some conditions, it directs some of the system low-priority and long running system functions.

SCPF typically is the last OS/400 job active when powering down the system, receiving control when the QSYSARB job ends.

11.1.2 System Arbiter Jobs (QSYSARB, QSYSARB2 through QSYSARB5)

The system arbiter (QSYSARB and QSYSARB2 through QSYSARB5) is started by SCPF and provides the environment for running high priority system functions, including:

- Subsystem monitor start, end

- Tracking ″restricted″ and ″non-restricted″ system status

- Device description object creation and deletion
- Device vary on and vary off locking
- Switched line connection processing
- General system ″even handlers″
- Sets up job management work areas based on system values such as QACTJOB and QTOTJOB

QSYSARB runs until the system is powered down. It is the highest priority OS/400 job and initially starts up the QLUS system job.

11.1.3 System Logical Unit Services Job (QLUS)

This system job is always started by QSYSARB. QLUS functions include:

- General management of communication devices, such as APPC devices
- Initial processing of received program start requests and routing to the appropriate subsystem monitor
- LU6.2 session management at vary on, vary off, for the Change Session Maximum (CHGSSNMAX) command function, and for ending an APPC conversation

For a single Performance Monitor time interval, CPU utilization can be 2% to 10% if a large number of APPC device descriptions are varied on during the same short period of time.

If QLUS CPU utilization is greater than 10% for some performance monitor interval or multiple time intervals, there is usually an APPC application design problem. One known cause is the short time span ending and starting of hundreds of APPC conversations that includes the APPC verb equivalent to the Change Session Maximum command function.

See Section 12.10, “APPC Programming Tips and Techniques” on page 289 for more detailed information in this area.

11.1.4 System Work Control Block Table Cleanup Job (QWCBTCLNUP)

This system job begins running toward the end of system IPL and typically ends before IPL is complete, but may still be running after IPL. QWCBCLNUP ″cleans up″ Work Control Block Table (WCBT) entries that have not yet been cleaned up prior to this IPL.

11.1.5 System Performance Adjustment Job (QPFRADJ)

This system job uses system value QPFRADJ to automatically monitor *shared pool* activity levels and page faulting rates. The QPFRADJ adjusts the storage pool sizes and pool activity levels to stay within ″good″ page fault guidelines for each pool.

System value QPFRADJ controls when the QPFRADJ system job does its pool management: (0 = no adjustment; 1 = perform adjustment at IPL; 2= perform adjustment at IPL and dynamically during normal system operation; 3 = perform adjustment dynamically during normal system operation).

Note: In some cases, QPFRADJ at IPL should not be used as IPL tuning may ″undo″ acceptable performance achieved by dynamic tuning or manual tuning done during normal system operation. In a ″steady state″ environment (same number of jobs active, no new applications), QPFRADJ is set to 0 after it has been used to obtain optimal paging rates across the system.

The following list contains some of the features of the QAFRADJ job:

- The tuner looks at the specific environment to determine an acceptable page faulting rate for each pool. Each pool is ″graded″ based on its priority, faulting rate, and the current size of the pool. The pools with the highest ″grades″ are allowed to take storage from pools with the lowest ″grades″. The tuner tries to balance the paging across pools evenly. There are 10 possible grades a pool can be assigned.
- The tuner allows most of the storage from inactive pools to be given to active pools (down to a minimum of 256 bytes).
- *BASE pool gets 30% of the storage available at IPL time once the machine and spooled pools are calculated. *INTERACT gets all of the remaining storage.
- The activity level is lowered only if paging is too high and no storage is available. Otherwise, the activity level is raised if there are any ineligible transitions.
- When the system is idle, the machine pool is **not** reduced. Also, the machine pool has a minimum value equal to 1.5-2.0 times the reserved size of the machine pool (depending on the configuration).

11.1.6 Database System Cross Reference Job (QDBSRVXR)

QDBSRVXR is a job that runs program QDBXREF at priority 0. QDBXREF updates the system database cross reference files with information enqueued by either system or user-initiated database requests, or generic object functions affecting database file objects (for example, RNMOBJ and CRTDUPOBJ). Information about relational database directories (WRKRDBDIRE) and SQL package objects including Distributed Relational Database (DRDB) information is also maintained by this job.

11.1.7 Database Server Jobs (QDBSRV01... QDBSRVnn)

These database server jobs run at priority 9 and perform data access path recovery during IPL and during normal system operation when necessary. These jobs are started at every IPL and remain active until the system is powered down. They remain active waiting for any recovery work the system determines to be necessary.

In V3R6, there is one of these jobs per processor. These jobs perform some System Managed Access Path Protection (SMAPP) work. For more information on SMAPP, see “Consider System Managed Access Path Protection (SMAPP)” on page 256.

Note: As part of the conversion of database reference information to Version 3 format, a QDBSRVnn job identifies files/table data to be converted by the QDBSRVXR job. Soon after the installation of the new release completes, one of the low priority QDBSRVnn system jobs runs program QDBXREFR to enqueue the work for QDBSRVXR to both refresh data in the old cross-reference database files and add data to the new cross-reference database files. After the work completes, both the additional fields and the new files reflect the current information from all of the database files on the system. All database files that exist in a library that existed during the installation are eventually processed by this new cross-reference conversion function.

This conversion may continue across IPLs until it completes processing all of the libraries, and it **waits forever** to obtain any lock before continuing with the next database file or library. Because of the locking implementation, applications that lock libraries or database files immediately after an IPL and continue to hold the locks until the machine is ended to a restricted-state may cause long delays in the post-install updates and additions to the cross-reference data.

See “V3R6 One-Time Cross Reference File Conversion” on page 258 for additional information.

The cross-reference conversion process also has system disk storage size considerations. As a guideline, a description of cross-reference information conversion is contained in the V3R1 ″information only″ APAR II08311.

11.1.8 Decompress System Object Jobs (QDCPOBJ1... QDCPOBJn)

These jobs decompress or compress system objects as needed and are started as part of IPL.

11.1.9 System Job Scheduler Job (QJOBSCD)

This system job performs the OS/400 job scheduling functions as specified by the Add Job Scheduler Entry (ADDJOBSCDE) command and Change Job Scheduler Entry (CHGJOBSCDE) command.

QJOBSCD is started by the system as part of the IPL and monitors time, date, and scheduled job parameters based on the job scheduler entries.

11.1.10 System Spool Maintenance Job (QSPLMAINT)

This job is responsible for spooled output management. QSPLMAINT is started at IPL and remains ″active″ during normal operations. It shows various ″status values″ based upon the function it is performing (deleting no longer needs spooled output, reclaims space from spooled data that has been unused or empty longer than the time period specified in system value QRCLSPLSTG) and performs spooled cleanup as part of the IPL after an abnormal system termination.

11.1.11 System Alert Manager Job (QALERT)

This system job performs the work necessary to process SNA alert support on the local system. This includes processing alerts received from remote systems and generating SNA alerts on the local system according to the appropriate message description alert parameters.

QALERT is started during an IPL and remains active until system shutdown.

11.1.12 System LU6.2 Resynchronization Job (QLUR)

This system job performs any *two-phase commit* processing.

QLUR is started during an IPL and remains active until the system is shutdown.

11.1.13 System File System Job (QFILESYS1)

The file system (QFILESYS1) system job supports the background processing of the file system. It makes sure that changes to files are written to disk. It also performs several general file system cleanup activities.

QFILESYS1 is started during an IPL and remains active until the system is shutdown.

11.1.14 Remote File System Communications Job (Q400FILSVR)

The remote file system communications job performs the Common Programming Interface (CPI) Communications (APPN or APPC) for the Remote File System. This job is started at every IPL.

11.2 V3R6 Subsystem Monitors

Several subsystem monitors are provided under OS/400. IBM licensed program applications may either place jobs into one of these subsystems or supply their own subsystems (such as ADSM/400).

This section gives a brief overview of key subsystem monitors. For applications, you must review application specific documentation to determine their unique subsystem monitor requirements.

IBM-supplied subsystems may use shipped system values (such as QCTLSBSD) and subsystem description parameters to determine the work assigned to a particular subsystem. This section describes the typical assignments. See *OS/400 Work Management Version 3*, SC41-4306 for detailed information.

Many of the IBM-supplied subsystem monitors have routing entries predefined for certain functions. Through manipulation of subsystem monitor description routing entries, class descriptions, and communication entries, the user may route IBM-provided applications to any subsystem and control the storage pool assignment and job priority within a subsystem. See index entries for additional information on assigning work to subsystem monitors and controlling user and IBM-supplied applications storage pool and run priority assignment.

For information on communication entries, routing entries, class descriptions, and auto start job entries of subsystem descriptions supplied as part of OS/400, refer to the *OS/400 Work Management Version 3*, SC41-4306.

Some of the subsystems included here have IBM-supplied job names for IBM-supplied functions, such as SNADS jobs. User-profile-name/job name are shown when the user profiles for ″IBM-supplied functions″ are defined by the function. This information may assist in collecting performance data for all the jobs related to a specific function.

11.2.1 QBASE subsystem

This subsystem can be the controlling subsystem on the AS/400 system (system value QCTLSBSD(QBASE) is specified) and is typically used in simple, single application environments. Running non-interactive and interactive applications in QBASE at the same time generally results in page faulting rates that approach the poor range.

The system is shipped with QCTLSBSD(QBASE) specified.

This redbook assumes a more sophisticated environment than can be supported with almost all work being performed in QBASE. For the remainder of this redbook, we assume QCTLSBSD(QCTL) is specified and at least the following subsystems are active and QBASE is not:

- QCTL
- QBATCH
- QINTER

Other IBM subsystems, such as QSPL, QSNADS, and so on, and user-defined subsystems may also be active.

System value QCTLSBSD specifies whether QBASE, QCTL, or some user-defined subsystem is the controlling subsystem.

System value QSTRUPPGM specifies the startup program that is called soon after the controlling subsystem has started at the end of IPL. The startup program starts the subsystem monitors that both IBM and the customer need to be active after IPL. As shipped from IBM, the startup program starts subsystem monitors QBATCH, QINTER, QSPL, QSNADS, QSERVER, and QCMN.

The user can use the Retrieve CL Source (RTVCLSRC) program to determine the IBM subsystems automatically started. Optionally, the user can modify this program, including adding their own subsystem monitors to start.

11.2.2 QCTL, QBATCH, QINTER Subsystems

QCTL is the default controlling subsystem. It is the only subsystem active when the system is operating in a *restricted state.* The console device jobs default to run in QCTL. In some system environments, QCTL may also do interactive and batch user application work. The Performance Monitor defaults to running in QCTL. System cleanup jobs, such as QSYSSCD set up under Operational Assistant, run in QCTL.

Most OS/400 commands that perform the Submit Job (SBMJOB) command default to placing the job request on job queue QBATCH that is assigned to subsystem QBATCH. This means, by default, typical *non-interactive jobs* run in subsystem QBATCH.

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

11.2.3 QCMN Subsystem

QCMN is shipped to make it easier to configure the setup for receiving remote program start requests that start up communication-based jobs on the AS/400 system.

As shipped, **all** communications-based jobs run in QCMN. This includes incoming user-written application programs, the APPC target session for display station sessions, target DDM sessions, and SNADS receiver jobs.

As shipped, the QCMN subsystem contains routing entries for Client Access/400 jobs. See Section 13.4.1, “Performance Tips/Techniques for Client Access/400 File Serving” on page 332 for more information on the Client Access/400 use of subsystem QCMN.

See also Section 13.9.2, “ADSM/400 Work Management” on page 367 for information on the ADSTAR Distributed Storage Manager/400 (ADSM/400) use of subsystem QCMN.

11.2.4 QSPL Subsystem

This subsystem is shipped to control all spooled output work, such as the ″Start Printer Writer (STRPRTWTR) command jobs.

11.2.5 QSNADS Subsystem

This subsystem performs document transmission, change management (SystemView System Manager/400 and SystemView Managed System Services/400) transmission, OS/400 Object Distribution Facility (ODF), and TCP/IP Simple Mail Transfer Protocol (SMTP) work.

There are several ″routing jobs active″ and a job for each ″send distribution″ defined for a remote location.

- QSNADS/QDIAnnnnnn jobs:

  These jobs perform Document Interchange Architecture (DIA) functions such as routing local system distributions, routing ″independent user″ functions, and routing local system host printing functions.

- QSNADS/QNFTP:

  This job performs most ODF send and receive functions.

- QSNADS/QROUTER:

  This job provides SNADS routing for *change management* functions. However, routing functions for normal SNADS distributions (for example, documents and ODF) are moved to the QMSF job (or jobs) that run in subsystem QSYSWRK.

- QSNADS/QZDSTART:

  This job is an auto start job when subsystem QSNADS is started. It starts the QDIA jobs, the QROUTER job, and the jobs for each remote system defined in the SNADS configuration *distribution queues*.

- QSNADS/remote-location-name:

  These jobs are the SNADS send jobs for each distribution queue defined in SNADS configuration distribution queues.

- QSRVBAS/QESTP:

  This job is activated as part of standard OS/400 support for receiving PTFs from either IBM or a customer service provider.

- QGATE/remote-location-name:

  These jobs are the SNADS send jobs for either SNADS bridge support such as for the MVS/VM bridge or change management distribution jobs when Managed System Services/400 or System Manager/400 are started for the local system.

If you already had QSNADS/remote-location-jobs active, you also have a corresponding QGATE/remote-location-job active when System Manager/400 or Managed System Services/400 is active.

- QGATE/TCPIPLOC:

  This job is activated when TCP/IP Simple Mail Transfer Protocol (SMTP) is activated for the local system.

11.2.6 QSYSWRK Subsystem

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

Subsystem description QSYSWRK is shipped with the **BASE* storage pool and is not included in the system-supplied start program QSTRUP. QSYSWRK is started by the SCPF job during IPL unless the system is IPLing to the restricted state. Subsequent topics within this chapter discuss the advantages that separate main storage pools for any subsystem may provide in improving overall system performance.

The following V3R6 ″facilities″ cause jobs to be to run in subsystem QSYSWRK by default. The user profile name and job name (user-profile-name/job name) are shown when the user profiles cannot be varied by the user.

- ManageWare/400 jobs:

  For information on ManageWare/400 jobs, refer to the *ManageWare/400 Administrator′s Guide*, SH19-4507.

- Directory Shadowing support job (QDOC/QDIRSHDCTL):

  If defined, this job keeps distribution directories updated (shadowed) across the defined systems. For information on directory shadowing, refer to *SNA Distribution Services - Version 3*, SC41-3410.

- LAN Server/400 File Server I/O Processor monitor job:

  There is one job active for each File Server I/O Processor varied on. The monitor job has the name of the **network server description** started for the File Server I/O Processor.

  For more work management information on the File Server I/O Processor support, see *LAN Server/400: A Guide to Using the AS/400 as a File Server*, GG24-4378. Performance considerations are discussed in this redbook and also in Section 13.5.2, “LAN Server/400 Tips and Analysis” on page 338.

- Operation Control Center/400 (SystemView System Manager/400 and SystemView Managed System Services/400):

  For a system defined as a *service requester*, job QSVSM\QECS is started.

  The following jobs that provide *change management* support under SystemView Managed System Services/400 support (managed site) or SystemView System Manager/400 support (manager site) may be active.

  For information on SystemView System Manager/400, refer to *SystemView System Manager/400 Use V3R6*, SC41-4321. For information on SystemView

Managed System Services/400, refer to *SystemView Managed System Services/400 Use V3R6*, SC41-4323.

- QSVMSS/QCQEPMON:

  This job monitors Managed System Services/400 work, including:

  - Completion of CL input streams run as a result of change request activities requested by the central site manager (such as V3R6 SystemView System Manager/400, NetView Distribution Manager V1R5 or later, and so on).
  - Scheduled jobs under change management support.
  - Notifying the central site manager that a scheduled job has completed.

- QSVMSS/QCQRCVDS:

  This job receives change management distributions from subsystem QSNADS jobs.

- QSVMSS/QVARRCV:

  This job accepts any remote command change request activities received from the central site manager.

- QSVMSS/QCQSVSRV:

  This job processes change request activities received from the central site. There may be multiples of these jobs. You can control the number of these jobs concurrently active by changing job queue entry QNMSVQ.

- QSVSM/QCQROMGR:

  This job sends remote commands to managed sites under V3R6 SystemView System Manager/400 if the Start Manager Services (STRMGRSRV) command has been issued on this local system.

- QSVSM/QNSCRMON:

  This job monitors the change management requests and initiates sending these requests to the managed system. This job should be active only if V3R6 SystemView System Manager/400 STRMGRSRV command has been issued.

- Mail Server Framework (QMSF/QMSF):

  There are one or more mail server framework jobs. Typically, there is only one job. The Start Mail Server Framework Job (STRMSF) command can be used to startup multiple QMSF jobs. Multiple QMSF jobs may improve performance during periods of excessive sending and receiving mail or SNADS distributions.

- TCP/IP support:

  In V3R6, TCP/IP support is part of OS/400. When the Start TCP/IP (STRTCP) command is issued, several jobs are started in QSYSWRK.

  For more information on the following TCP/IP jobs, refer to *TCP/IP Configuration and Reference - Version 3*, SC41-3420.

  - TCP/IP Main Job(QTCP/QTCPIP)
  - TCP/IP File Transfer Protocol (FTP) Server (QTCP/QTFTPxxxxx):

    There may be more than one active.

  - TCP/IP TELNET Server (QTCP/QTGTELNETS):

    There may be more than one active. Multiple TELNET sessions (typically 20) are managed by a single QTGTTELNETS job.

  - Simple Mail Transfer Protocol (SMTP) client (QTCP/QTSMTPCLNT)
  - Simple Mail Transfer Protocol (SMTP) server (QTCP/QTSMTPSVSR)
  - Simple Mail Transfer Protocol (SMTP) bridge client (QTCP/QTSMTPBRCL)
  - Simple Mail Transfer Protocol (SMTP) bridge server (QTCP/QTSMTPBRSR)
  - Simple Network Management Protocol (SNMP) server (QTCP/QTSNMP)
  - Simple Network Management Protocol (SNMP) server (QTCP/QTMSNMPRCV)
  - Simple Network Management Protocol (SNMP) server (QTCP/QSNMPSAV)
  - Line Printer Daemon (LPD) server (QTCP/QTLPDxxxxx):

    There may be more than one active.

  - APPC over TCP/IP if AnyNet support is in use (QTCP/QAPPCTCP):

    AnyNet support is part of V3R6 OS/400 and, if configured, supports APPC data over TCP/IP and TCP/IP data over APPC. The network attributes must specify to allow AnyNet support (ALWANYNET(*YES)).

Subsystem QSYSWRK is shipped with several *auto start job entries*, including QSYSWRKJOB, QFSIOPJOB, and QZMFEJOB. These jobs run at the start of subsystem QSYSWRK and restart LAN Server/400 jobs, other QSYSWRK processing, and mail framework jobs that are previously listed. Once the normal production mode jobs are active, these auto started jobs end normally.

If you do ENDSBS QSYSWRK *IMMED (immediate), all jobs abnormally end and cause some system overhead in generating job logs. Re-issuing STRSBS QSYSWRK automatically restarts all of the jobs discussed previously except the TCP/IP jobs. You must do ENDTCP and follow with STRTCP to make TCP/IP support operational again.

11.2.7 QLPINSTALL

This subsystem performs Licensed Program (LP) installation functions.

11.2.8 QPGMR

This subsystem is available for application development functions.

11.2.9 QSERVER

This subsystem is shipped with V3R6 OS/400 and runs the host server jobs for Client Access/400 *file serving* and *database serving* functions. There is one autostart job and one file server job for each active client and one database server job for an active database serving session as shown:

- USERID/QPWFSERV:

  File serving support includes storing programs and files as a network drive (″virtual disk″) for the attached client.

- QPGMR/QSERVER:

  This auto start job sets up the file serving and database serving environment on the AS/400 system.

- QUSER/QZDAINIT:

  There is one of these database serving (remote ODBC and SQL) functions for each active client session. QZDAINIT is implemented as a pre-started job.

11.2.10 QADSM

This subsystem is active when ADSTAR Distributed Storage Manager/400 (ADSM/400) is installed and active. Most ADSM functions run and are controlled by jobs in this subsystem. Actual send and receive functions are performed in QSYSWORK for TCP/IP and QCMN for APPC connections.

For additional information on ADSM/400, refer to Section 13.9.2, “ADSM/400 Work Management” on page 367.

11.2.11 Q1ABRMNET

This subsystem is active when Backup Recovery and Media Services/400 is active. For additional information on ″BRMS/400″, refer to product documentation and the redbook *Backup Recovery and Media Services/400 Implementation Tips and Techniques*, GG24-4300.

11.2.12 QLANRES

QLANRES subsystem is active when Local Area Network Resource Extension and Resources/400 is active. The LANRES jobs in QLANRES perform the communications input and output with the Novell NetWare server.

LANRES functions such as file serving and print serving run as submitted jobs to any subsystem designated by the LANRES administrator.

For additional performance information, refer to Section 13.7, “LANRES/400 Performance Tips” on page 361.

11.2.13 QAUTOMON

The QAUTOMON subsystem is provided with SystemView OMEGAMON Services/400, program number 5716-ES1. QAUTOMON runs the OMEGAMON ″condition″ monitoring jobs and the jobs for the AUTOMATED FACILITIES/400 feature of 5716-ES1.

The SystemView OMEGAMON Services/400 OMEGAVIEW/400 feature is a client application running on a personal computer under OS/2. OMEGAVIEW/400 provides a graphic interface for defining automated policies used by AUTOMATED FACILITIES/400 and graphically displaying the conditions on AS/400 systems monitored with the base OMEGAMON/400 support.

OMEGAVIEW/400 communications jobs that communicate with cooperative programs on the personal computer default to running in subsystem QCMN.

For a complete description of OMEGAMON Services/400, its features, and performance tips, refer to product documentation and the redbook *Managing Operations on AS/400s with IBM SAA SystemView OMEGAMON Services/400*, GG24-4136. Some performance tips are included in this redbook in Section

12.11, “SystemView OMEGAMON Services/400 Tips and Techniques” on page 293.

The following jobs are an example of the OMEGAMON support jobs that run in subsystem QAUTOMON:

- QAUTOMON/OM4DS
- QAUTOMON/OM4DSEV
- QAUTOMON/OM4DSEVL
- QAUTOMON/OM4DSNET
- QAUTOMON/OM4LLBD
- QAUTOMON/OM4SM
- QAUTOMON/OM4SMSQ
- QAUTOMON/Q1CCOMSRV
- QAUTOMON/Q1CDTMSNAH
- QAUTOMON/Q1CDTMSNAP

  There may be multiples of these jobs.

11.3 Review System Values

There are some system values that affect performance such as QMCHPOOL, QPFRADJ, QACTJOB, QTOTJOB, QDEVRCYACN, and QCMNRCYLMT. Review these values because they can relate to your situation.

Some key considerations for these system values include:

- QCTLSBSD: All examples given in this section assume that QCTLSBSD is set to QCTL (or a user-defined SBSD). If this system value is QBASE, change it to QCTL and re-IPL your machine.

- QMCHPOOL: Set the QMCHPOOL size according to *OS/400 Work Management Version 3*, SC41-4306, algorithms but we suggest you increase the value by 10%. This value directly controls the amount of memory reserved for the system′s own workload. If the QMCHPOOL is set too small, you notice a high fault rate in pool 1 and this severely impacts all jobs throughout the system.

  See Section 3.9.5, “Working Set Size” on page 39 for specific V3R6 machine pool tuning details.

- QPFRADJ: Set QPFRADJ to 3 unless you are satisfied with your own tuning of pool activity levels and storage sizes. This is discussed in more detail later in this chapter.

  If you are satisfied with pool activity levels and storage pool sizes, turn automatic tuning off by setting QPFRADJ to 0.

- QTSEPOOL: This value is the time slice end pool. This specifies whether interactive jobs should be moved to another main storage pool when they reach time slice end. The job is moved back to the pool it was originally running in when a long wait occurs. This may help minimize the effect on interactive response time of other interactive jobs when one interactive job is performing a long running function. If you allowed the IPL performance adjustment to tune the system, you should set this value to *BASE.

- QACTJOB: This value controls the initial number of *active* jobs for which auxiliary storage is to be allocated during IPL. Use the value found in the upper right corner of the WRKACTJOB display at the busiest time of day and use this value for system value QACTJOB.

Note: Please remember that the change becomes active only after the next IPL.

If the value of QACTJOB is reached during normal operation, the system allocates auxiliary storage for the number of jobs specified by system value QADLACTJ. QADLACTJ is shipped with a value of 10 which is the recommended value. Setting QADLACTJ closer to 1 can cause system overhead if there is a sudden increase in the number of active jobs. Setting QADLACTJ to a value close to 100 also causes overhead as space for 100 new jobs is created as soon as there is one active job over the QACTJOB value.

Note: You need to keep QACTJOB, QTOTJOB, QADLACTJ, and QADLTOTJ at reasonable values. If you make QACTJOB and QTOTJOB excessively high, the IPL is slowed due to excessive storage allocation. If you make QACTJOB and QTOTJOB too small for your environment and you make QADLTOTJ and QADLACTJ excessively large, run-time performance is impacted.

See the following description for QTOTJOB and QADLTOTJ for more information on these variables.

- QTOTJOB: This value controls the total number of jobs for which the auxiliary storage is to be allocated during IPL. Jobs in job queues, active jobs, and jobs having output on output queues are included in this value.

  The number of jobs in the left-hand column of the WRKSYSSTS display plus 20% may be used to set the system value QTOTJOB but be aware of the following conditions.

  Remember to clear output queues regularly because OS/400 reserves auxiliary storage for a job even though the job is inactive as long as there is at least one spooled output file for that job. The more files there are in output queues, the more jobs there are on the WRKSYSSTS display.

  Note: If you have a high number of spooled files when using WRKSYSSTS and add 20% more to set the QTOTJOB value, this significantly adds to the IPL time. Performance is also degraded at run time for any system functions that search through the system-wide Work Control Block Table (WCBT). These functions include WRKACTJOB, WRKJOB, and STRSBS.

  If the number of jobs in the system reaches the QTOTJOB value, OS/400 allocates auxiliary storage for jobs according to the QADLTOTJ value. This allocation is done while there are active jobs running so users may notice a slight degradation in response times. It is better to set the QTOTJOB value high enough not to be reached during the day.

  Note: Please remember that any change is active only after the next IPL.

- QRCLSPLSTG: When a spooled file is printed or deleted from a spooled output queue, or an entire spooled output queue is cleared, the space occupied by the files (database members) remains allocated to spool for reuse. To return this space to the system, OS/400 provides two interfaces, the Reclaim Spool Storage (RCLSPLSTG) command and the system value QRCLSPLSTG. QRCLSPLSTG specifies a ″number of days to keep printed or deleted spooled file storage allocated″.

  *NONE may also be specified for QRCLSPLSTG. ***NONE should not be used.** *NONE means space is de-allocated immediately and given back to ASP free space. *NONE can cause poor performance since the system cannot re-use existing spooled space and must create a new database member and allocate new storage each time a new spooled file is opened by the system. This create and allocate space overhead slows down the affected job, and possible the entire system.

You should also consider using the AS/400 Operational Assistant Menu (GO ASSIST) options to cleanup system spooled file output, such as job log and problem dump files.

- QCMNRCYLMT: The recommended value is (2 5), which is a maximum of two additional sets of communication lines or control unit retries within a five-minute interval. Never set the first value (additional retries) equal to or greater than the second (time interval) value, excluding (0 0).

  QCMNRCYLMT(0 0) means that no additional sets or ″second-level″ line or control unit error recovery attempts should be performed. The (0 0) value should be used only in rare instances when a unique communication environment has determined (0 0) to be a requirement.

  Note that each line and controller description object has a ″second-level″ recovery parameter (CMNRCYLMT) that normally defaults to (2 5). This CMNRCYLMT parameter has the same meaning as discussed for the system value QCMNRCYLMT. CMNRCYLMT can be set to use QCMNRCYLMT if CMNRCYLMT(*SYSVAL) is specified. Normally, any changes from a (2 5) value are made for a specific line or control unit.

  Incorrect setting of a QCMNRCYLMT/CMNRCYLMT value causes the system to perform this ″second-level″ recovery continuously if the time interval expires before the additional sets of retries are completed and the failing line or control unit remains in an error state. Under some conditions, the continuous retries consume significant system resources. If this occurs, stop the process by varying the object off.

- QDEVRCYACN: The default for this value is *MSG, which results in an error message to an active workstation job or communication target job if the device being used has a failure. Many applications do not check for workstation or communication device failures and may loop when the failure goes undetected. For workstations, the system detects successive error messages and abnormally ends the job. However, while a large number of active jobs are looping, overall system performance may degrade. Consider adding error handling to the programs or setting QDEVRCYACN to *ENDJOB or *ENDJOBNOLIST. This abnormally ends the job with minimum impact to overall system performance. *DSCMSG (disconnect the job, and signal an error if the job is reactivated) has the least system impact. If this technique is used, those responsible for monitoring normal system status must understand this function is being used. Once the reactivation occurs, the program or job has the same error detection considerations for QDEVRCYACN(*MSG).

- Use system value QDSCJOBITV for controlling how long disconnected jobs remain in disconnected status. When the QDSCJOBITV time limit is reached, the system automatically ends the disconnected jobs.

- Examine the system value QCTLSBSD:

  The system value QCTLSBSD describes the name of the first subsystem to be started after IPL. If you only want to use one subsystem (QBASE), the value should be ″QBASE QSYS″; normally, value ″QCTL QSYS″ is used which means that you have subsystems QBATCH, QCMN, QCTL, QINTER, QSNADS, and QSPL started automatically.

11.4 Consider Dividing the Main Storage into Separate Pools

Most AS/400 environments run interactive, batch, and spooled jobs. Certain AS/400 environments also run SNADS, Client Access/400, or TCP/IP support. There are some production environments where running different job types in separate storage pools can minimize the impact of certain jobs on other jobs. Using separate storage pools is more feasible when the system has a large amount of main memory.

The key is to partition some jobs into separate storage pools so that critical jobs are not affected by the activity of other jobs, including job start and termination. Using separate subsystems is the easiest way to control this, even though a single subsystem can use multiple routing entries and job queues to route jobs to separate pools.

11.4.1 Creating Separate Memory Pools for Subsystems

The next step is to check that all subsystem descriptions have at least two storage pools defined. As an example, we can check the QINTER subsystem description.

Enter the DSPSBSD QINTER command and choose option 2 for pool definitions; make sure that pool number 1 is *BASE and pool number 2 is *INTERACT. If there is only one pool defined, you must create another one using the CHGSBSD SBSD(QINTER) POOLS(2 *INTERACT) command.

Pool number 1 is the pool in which the subsystem monitor runs. To verify that all of the jobs in QINTER go to pool number 2, choose option 7 to display routing entries. If they point to pool number 1, enter the CHGRTGE SBSD(QINTER) SEQNBR(nnnn) POOLID(2) command for each routing entry. Replace ″nnnn″ with actual sequence numbers. CHGRTGE may be issued while the subsystem is active and affects jobs that start after the change has completed. After this is done, enter the WRKSHRPOOL command and verify that *INTERACT has a size and activity level. For example, if you have 48000KB of main storage and a pool of 10000KB, divide the pool by 1600 and the result is 6.2 which you round to 6. This means that there are only six jobs doing work in the memory pool at any given time and all of these jobs have 1666KB of memory to use. Refer to Table 38 on page 384 and to the *Work Management Guide* for more accurate values.

The reason for using a shared pool *INTERACT is that the automatic tuning adjusts shared pools only. You can use a private pool (pool that has only a size defined) for QINTER if you do not plan to use automatic tuning.

If you have multiple subsystems for interactive jobs, you should consider using shared memory pools. A shared pool is usable by more than one subsystem and is identified as *SHRPOOLn where ″n″ is between 1 and 10. This is useful if common applications are run from workstations attached to different subsystems.

The next subsystem description to be checked is QSPL, which is the subsystem where all of the printer output is produced. Enter the DSPSBSD QSPL command and choose option 2 to verify pool definitions and option 7 to verify that routing entries direct the jobs to pool number 2. After this is done, enter the WRKSHRPOOL command to verify that pool *SPOOL has memory size and activity level. A reasonable amount is 350KB and activity level 5 without *IPDS printers and approximately 2000KB if you use *IPDS printers. If you make extensive use of AFP overlays and segments, consider making the storage pool 4000KB (4MB). For more information about the size and activity level settings of *SPOOL, refer to the *OS/400 Work Management Version 3*, SC41-4306. Refer to Table 41 on page 386 for more accurate values to use.

When you are creating your own subsystems, it is a good idea to create two storage pools (each one in *BASE). All routing entries should point to storage pool number 2. If you later decide to separate jobs in these subsystems from *BASE, you can change the subsystem pool definitions while the subsystem is active.

You should always consider separate storage pools for unique applications, such Client Access/400 host servers, TCP/IP functions, ADSTAR Distributed Storage Management/400 (ADSM/400). See index entries for these applications for additional details.

If you have sufficient main storage and understand the program or files used in your various applications, you should also consider using the SETOBJACC command and expert cache support. See Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413 for additional information.

11.5 Additional Considerations

Here are a number of additional considerations which often are beneficial to performance.

11.5.1 Separate Subsystems

The following two examples are separate subsystems:

- Have ″critical″ batch jobs run in one subsystem and unplanned or ″ad hoc″ batch type functions run in a separate subsystem. Separate job queues for each subsystem help ensure an ″ad hoc″ job does not get in between two critical, long-running jobs.
- Consider grouping interactive jobs into separate subsystems based on their connectivity to the system or association with separate applications. One example is to have local twinaxial or ASCII workstation jobs in one subsystem and all other interactive jobs in another subsystem. This second set of jobs are those on LAN lines or remote communication lines.

The separate interactive subsystems prove useful if, for example, a communications line becomes unusable or if a new site is added. The recovery procedures done for the remote users do not affect the throughput of the local workstation users. You should also consider placing batch jobs in a separate pool during the day because batch jobs normally take more system resources than interactive jobs.

11.5.2 Separating QBATCH from *BASE

An important factor of improving the performance is to separate batch jobs from all of the other jobs running on the AS/400 system.

To estimate the amount of memory needed by a job, bear in mind that one query requires about 2MB of memory and one compile may require 12MB or more of memory. In this example, if you intend to compile programs and run queries during the day, the initial amount of storage for *SHRPOOL1 is 14MB and the activity level is 2. Please refer to Table 38 on page 384 for the accurate values to use.

If most of your batch jobs are run at night, you might want to make sure that they are ready in the morning when people come to work. The trick to use is that in the evening, enter CHGSBSD SBSD(QBATCH) POOLS(1 *BASE) and after that, end all of the subsystems not needed during the night. Then QBATCH has more memory and the job throughput should be better.

11.5.3 Batch-Like Jobs

There are many jobs that should be run in a batch mode because of their heavy use of system resources. Good examples are save/restore operations, cleanup jobs(CLROUTQ), query jobs, and so on.

The advantages of submitting a job in batch mode are:

- You do not have to wait for the job to end before regaining the use of your terminal.
- Your job does not steal resources from other interactive users running at the same priority.
- There are no sudden slowdowns of the response times.

The way to submit a job is through the SBMJOB command. For example, if you want to save a library during the day, enter the SBMJOB command and press PF4 for prompt. The command to run is SAVLIB. Enter that and press PF4 again for parameters. Enter parameters and press Enter to return to the initial prompt display and press Enter again to submit the job. You can monitor the status of your jobs with the WRKSBMJOB command or use the WRKUSRJOB command to monitor all of the batch jobs in the system no matter who submitted them. Do not get confused if job type CMNEVK appears on the display because it is a batch job started by the workstation function of Client Access/400.

11.5.4 Communications Subsystem

You should consider creating an additional memory pool for the communications subsystem QCMN if you have many Client Access jobs or if you use many line transmissions that seem to take an unusually long time to complete. The reason for a separate memory pool is that even though communications jobs need only a small amount of memory at a time, they must access the memory before a timeout limit is reached. Failing to do so leads to error recovery procedures that can use considerable amounts of system resource. One way to find information about timeouts is to start service tool (STRSST command) and search for communications errors in the error log. If you find messages that are not self-explanatory, contact IBM service for assistance.

11.6 How to Use System-Provided Tuning

The following section restates the recommendation to use the system-provided automatic tuning provided by system value QPFRADJ and the Performance Tools Advisor function *before* using the Performance Tools reports.

11.6.1 Automatic Tuning (QPFRADJ) Basic Operation

Setting system value QPFRADJ to value 2 or 3 starts up the QPFRADJ job. QPFRADJ uses less than 0.5% CPU every 20 seconds. It analyzes page faulting and Wait-to-Ineligible versus Active-to-Wait ratios for system and shared pools: *MACHINE, *BASE, *INTERACT, *SPOOL, and shared pools *SHRPOOL1-10.

Note: QPFRADJ does not adjust private pools. Based on conditions lasting over 80 seconds, QPFRADJ may change a shared pool′s activity level or move storage to one pool from another shared or system pool that has acceptable page faulting values. If there are several pools with high fault rates, remember that QPFRADJ gives first priority to pool 1 which is *MACHINE and second to *INTERACT pool.

Setting QPFRADJ to value 2 resets pools during each IPL in addition to tuning the system dynamically during normal operation. Review *OS/400 Work Management Version 3*, SC41-4306 for more details on real-time automatic system tuning. The difference between values 2 and 3 is that value 3 does not adjust the pools during IPL. The recommendation is that if you use automatic tuning, you should set the QPFRADJ value to 3.

If you are satisfied with the pool activity level and storage pool sizes and are in a reasonably stable operating environment, turn off QPFRADJ automatic tuning by setting QPFRADJ to 0. If there are mysterious slowdowns with the response times, check if the automatic adjustment is on. If it is, turn the adjustment off and calculate the pool size and activity level for subsystem QINTER based on the guidelines found in *OS/400 Work Management Version 3*, SC41-4306. For example, 60MB of memory and an activity level 20 is a good starting point for an AS/400 system with 128MB of memory. Please refer to Table 38 on page 384 for more accurate values.

11.6.2 Using QPFRADJ Externalized Parameters (V3R7 only)

In V3R7, the ″rules″ by which the automatic system tuning job QPFRADJ assigns storage to a shared pool in need of more storage have been externalized to the OS/400 user. A set of parameters for ***each shared pool*** can now be seen and changed through the Work with Shared Pools (WRKSHRPOOL) or the Change Shared Pool (CHGSHRPOOL) commands. These parameters include:

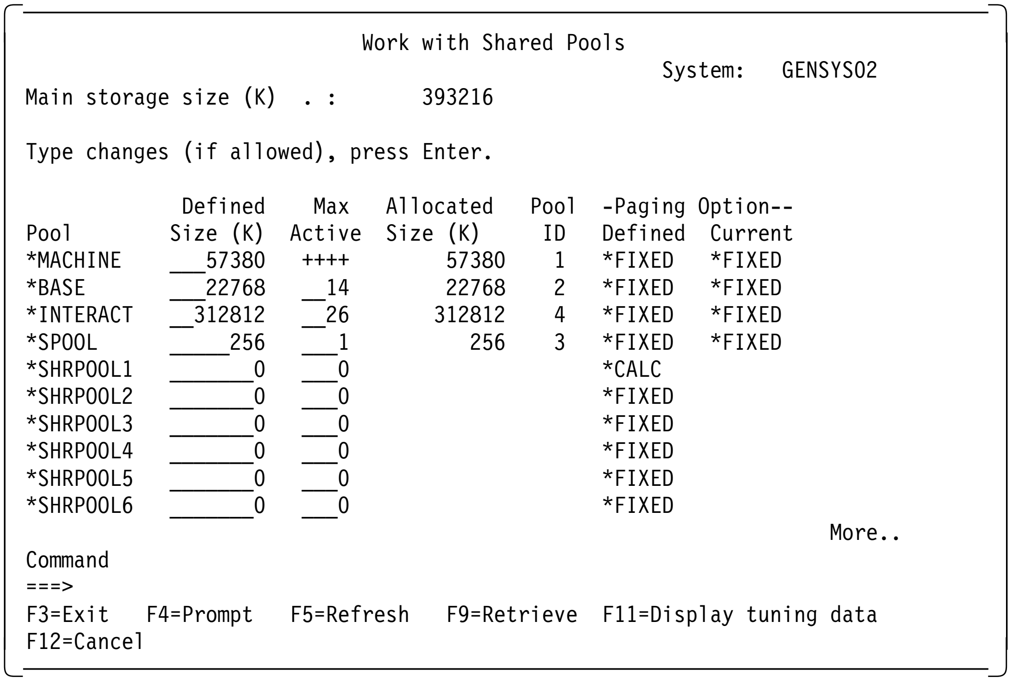
- Pool priority value
- Pool minimum and maximum storage size in percentage of total main storage
- Pool page faults per second range for each shared pool and jobs running in that pool

You may consider changing the storage pool minimum and maximum percentage to prevent an important storage pool from getting ″too small″ or ″too large″.

For example, you may have non-interactive jobs running at night that result in QPFRADJ shrinking a storage pool used for interactive work (for example, *INTERACT) to a small size. Unless the interactive pool is specifically made larger in the early morning hours, when interactive users sign-on in the morning, they may experience significantly slower response times since storage from a ″non-busy″ interactive job pool has been given to the pool used for nighttime non-interactive work.

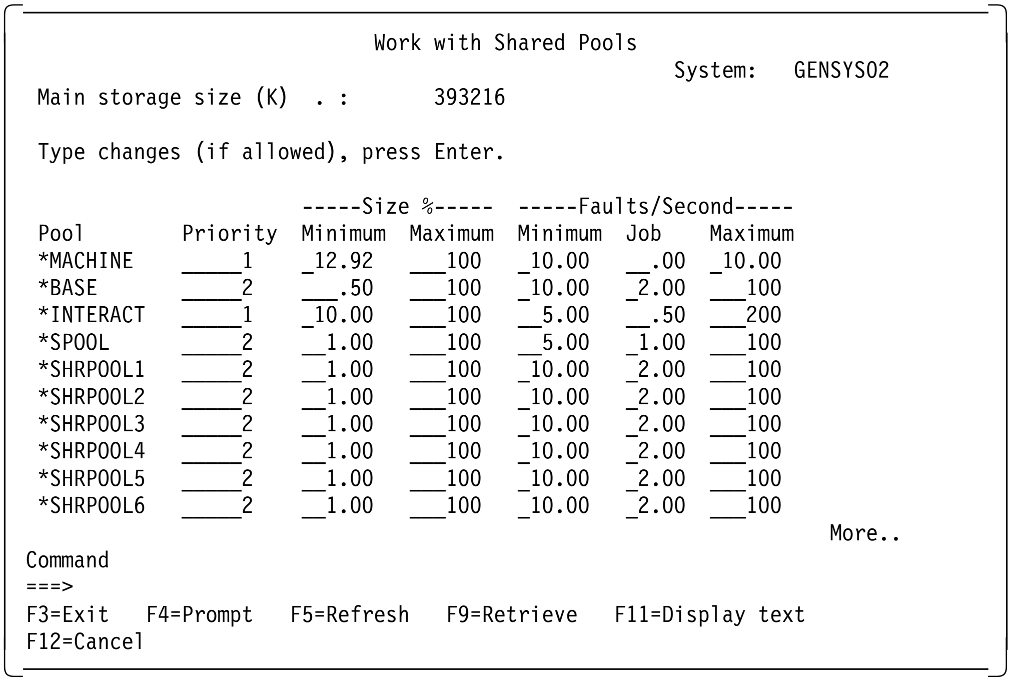
QPFRADJ detects the need for more interactive job pool storage but in many cases it may take ″too long″ to increase the pool size large enough to achieve expected interactive performance. By specifying a ″large enough″, non-default ***minimum storage %*** value, the customer can be guaranteed that the interactive storage pool has ″enough″ storage to begin daytime production mode with good performance.

The following displays show this new support. The first display is the same as V3R6, except for function key F11:



```
                            Work with Shared Pools
                                                        System:   GENSYS02
Main storage size (K)  . :       393216

Type changes (if allowed), press Enter.

             Defined    Max   Allocated   Pool  -Paging Option--
Pool         Size (K)  Active  Size (K)    ID   Defined  Current
*MACHINE     ___57380   ++++       57380    1   *FIXED   *FIXED
*BASE        ___22768   __14       22768    2   *FIXED   *FIXED
*INTERACT    __312812   __26      312812    4   *FIXED   *FIXED
*SPOOL       _____256   ___1         256    3   *FIXED   *FIXED
*SHRPOOL1    _______0   ___0                    *CALC
*SHRPOOL2    _______0   ___0                    *FIXED
*SHRPOOL3    _______0   ___0                    *FIXED
*SHRPOOL4    _______0   ___0                    *FIXED
*SHRPOOL5    _______0   ___0                    *FIXED
*SHRPOOL6    _______0   ___0                    *FIXED
                                                                  More..
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve  F11=Display tuning data
F12=Cancel
```

Use ***F11*** to see the new tuning data for QPFRADJ system job.



```
                            Work with Shared Pools
                                                        System:   GENSYS02
 Main storage size (K)  . :       393216

 Type changes (if allowed), press Enter.

                        -----Size %-----  -----Faults/Second-----
 Pool        Priority  Minimum  Maximum  Minimum  Job     Maximum
 *MACHINE    _____1    _12.92   ___100   _10.00   __.00   _10.00
 *BASE       _____2    ___.50   ___100   _10.00   _2.00   ___100
 *INTERACT   _____1    _10.00   ___100   __5.00   __.50   ___200
 *SPOOL      _____2    __1.00   ___100   __5.00   _1.00   ___100
 *SHRPOOL1   _____2    __1.00   ___100   _10.00   _2.00   ___100
 *SHRPOOL2   _____2    __1.00   ___100   _10.00   _2.00   ___100
 *SHRPOOL3   _____2    __1.00   ___100   _10.00   _2.00   ___100
 *SHRPOOL4   _____2    __1.00   ___100   _10.00   _2.00   ___100
 *SHRPOOL5   _____2    __1.00   ___100   _10.00   _2.00   ___100
 *SHRPOOL6   _____2    __1.00   ___100   _10.00   _2.00   ___100
                                                                  More..
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display text
F12=Cancel
```

Before changing any of these values, it is useful to understand what the column headings for Priority, ″Size %″, and ″Faults/Second″ actually mean. In general, the priority value has the greatest influence in determining if a storage pool should receive additional storage. Storage pool priority and the values shown under the other column headings are taken together for each pool to give a ″grade level″ for the storage pool. A pool with a better grade level is more likely to receive additional storage than a pool with a lower grade level.

Priority

The priority of this shared storage pool is relative to the priority of other shared storage pools. The valid range for priority is 1-14, where 1 is the best priority and 14 is the worst priority. This value is used by the system if the QPFRADJ system value is set to 2 or 3. Entering a value of *DFT causes the default value to be displayed. All default values are highlighted on the display.

As shipped with V3R7, the *MACHINE pool has the best priority value of **1**, compared to other shared storage pools.

On non-server models, pool *INTERACT is shipped also as priority 1 and pool *BASE has priority 2.

On server models, pool *INTERACT is shipped with priority 2 and pool *BASE has priority 1.

All other shared storage pools are shipped as priority 2.

Size Percentage

This is the percentage of total main storage to allocate to this storage pool. This value is used by the system if the QPFRADJ system value is set to 2 or 3. Entering a value of *DFT causes the default value to be displayed. All default values are highlighted.

- Minimum

  This is the minimum amount of storage to allocate to this storage pool (as a percentage of total main storage).

  On most systems, the machine pool has the highest default value for ″minimum percentage of total storage.″ For the machine pool, this value changes throughout the day and takes into account the minimum amount of storage that must be reserved for system use. This amount of storage can be seen on the WRKSYSSTS command output.

  On traditional V3R7 models, *INTERACT pool has the largest amount of minimum storage above the machine pool and defaults to 10% of available main storage. All other shared storage pools have a minimum of 1%. The *BASE storage pool is calculated using the system value QBASPOOL. For the WRKSHRPOOL example shown with system value QBASPOOL(2000) - 2MB and total main storage of 393216K, the minimum size for *BASE is calculated as follows:

  ```
  (2000/393216) x 100% = .50% (truncated to 2 decimal positions)
  ```

- Maximum

  This is the maximum amount of storage to allocate to this storage pool (as a percentage of total main storage). The actual maximum amount of storage that is assigned is determined by this percentage and the amount of storage allocated to the other active pools.

The system defaults to allowing all pools to grow to up to 100%, which rarely happens.

Faults per Seconds

This is the number of page faults per second for the system to use as a guideline for this pool. This value is used by the system if the QPFRADJ system value is set to 2 or 3 (automatic performance adjust). Entering a value of *DFT causes the default value to be displayed. All default values are highlighted on the display.

- Minimum

  This is the minimum page faults per second to use as a guideline for this storage pool.

- Job

  The page faults per second for each ″active″ job to use as a guideline for this storage pool. In this calculation, a job is counted as active if it used any CPU time in the last 20 seconds.

- Maximum

  This is the maximum page faults per second to use as a guideline for this storage pool.

Page Faults per Second Special Notes

- Machine Pool

  The minimum and maximum page fault rate for the Machine pool is based on ″Chapter 14″ of *OS/400 Work Management Version 3*, SC41-4306, where the value of ″Acceptable is < 10.″

- *INTERACT Pool

  The industry average transaction rate is about 1 fault every 20 seconds. We assume 10 page faults per transaction is acceptable because with today′s AS/400 disks, this only adds .1 second to interactive response time (assuming the disk response time is .01 second). Ten faults per transaction divided by 20 seconds per transaction equal 0.5 faults per job per second, as shown in the WRKSHRPOOL example, under the heading:

  ```
  -------- Size % --------     -------- Faults/Second ---------
  ```

  You can calculate your own value if you consider your transaction rate or page faults per transaction are different than the industry average.

- *BASE and *SHRPOOL1 to *SHRPOOL10 Pools

  We assume non-interactive jobs are run in these pools.

  Assuming .01 second per disk operation, a non-interactive job at 10 page faults per second has a disk utilization of .1. Since for a single job, the utilization of all resources must be 1.0, this leaves .9 left for the processor. That is, 90% of the time available is available for the processor, if needed.

  Ten becomes the minimum page faults per second for the storage pool and two becomes the job faults per second because in most cases, larger number of jobs are run on the faster processor speeds where customers do not want heavy page faulting for these jobs. Ten plus two faults per job gradually decreases the faulting (on a per job basis) that is tolerated.

- *SPOOL Pool

  The default values for the SPOOL pool are ″in-between″ interactive and non-interactive work.

Remember, a pool is determined to be in need of more storage if its page faults per second are beyond a threshold value. Also, a pool that needs more storage cannot receive that additional storage unless another storage pool can give it up without reaching its own threshold value page faults per second and minimum size.

Note: QPFRADJ does not make adjustments to its maximum page fault guidelines based on processor speed groups. This processor dependent set of ″good″ values are documented in *OS/400 Work Management Version 3*, SC41-4306, for each pool. QPFRADJ defaults to maximum per pool value of 100 for all non-machine pool pools and 200 for *INTERACT. The faster 530 and 53S models can tolerate higher page fault rates before page faults per second impact performance.

11.6.3 Performance Tools Advisor

The Advisor output is grouped under the following headings: Recommendations, Conclusions, and Interval Conclusions.

The Advisor makes changes to pool sizes and activity levels. These changes are not made dynamically, but only after the operator tells the Advisor to make the changes or to ignore the recommendations. The tuning is done by using F9 on the Display Recommendations display. Pool and activity level changes can be made to all of the pools on the system. The Advisor can analyze performance data collected from other systems, but when restored on the system for analysis, the changes are not be made if the data was collected on another system.

Another reason the Advisor does not make a change is if its analysis indicated that pool size changes were made while the Performance Monitor was active. Changes made by the QPFRADJ job while the monitor was active are an example of when a recommendation is not made.

Normally, it is recommended that QPFRADJ tuning be completed and system value QPFRADJ set to 0 before running the Performance Monitor and using the Advisor. The Advisor does not perform response time or job-level analysis, so the recommendation is to let the Advisor do as much as it can and then use the Performance Tools reports or other performance tools for further problem analysis and resolution. See the ″Chapter 5. Advisor″ chapter of the *Performance Tools/400 Version 3*, SC41-4340 for detailed information.

11.7 The Usage of Main Memory

Memory wait time is a key component of performance, but it cannot be directly measured with any of the performance tools. However, the effect of memory demand can be observed, measured, and controlled to a certain degree using the storage pools′ page fault rates. This observation can be done either interactively with the WRKSYSSTS command or by analyzing the System report generated using Performance Tools/400. Also, refresh the display at about two-minute to five-minute intervals, as the results are more reliable with longer time periods between refreshes.

Main storage requirements of a job depend, at a specific time, on the job′s size and the memory demand made by other concurrent jobs sharing the same main storage pool. The main storage requirement is affected by the program size, whether or not the program used by the job is also used by other jobs, and the amount of temporary storage in use (such as file buffers and program variables).

Paging activity in storage pools can be caused by either an OS/400 job or SLIC tasks. If the NDB page fault rate in the machine pool (pool 1 or *MACHINE) is greater than five to 10 faults/sec on a system, performance may be improved by increasing the size of the machine pool, thereby reducing the page fault rate. Remember that the only way of controlling the fault rate in the machine pool is by changing the size of the pool; you cannot increase or decrease the activity level of *MACHINE pool. Refer to the *OS/400 Work Management Version 3*, SC41-4306, *Performance Tools/400 Version 3*, SC41-4340, or Appendix A, "Guidelines for Interpreting Performance Data" on page 383 for more information on paging rates.

SLIC tasks that support workstation I/O run in the machine pool can become a bottleneck if the pool′s fault rate is too high. For example, faulty workstation cabling can increase the page fault rate. If there is lots of faulting in the *MACHINE pool, try starting the System Service Tool and check if there are several error log entries for local workstations. If adding memory to the machine pool does not reduce the page fault rate, do not continue to increase the size of the pool.

On systems that are performing well, the sum of database and non-database page faults in each storage pool is usually less than 50 to 200 faults/second, depending on the AS/400 model. Higher performing systems generally generate more page faults/second. Decide how to divide main storage between the pools based upon the fault rates and performance requirements for the jobs running in those pools. The commands to print data collected with the STRPFRMON command show the paging rates of individual jobs and storage pools.

For systems with limited main storage sizes, analyze job PAGs to see if savings can be made. Opening and closing *seldom* used files each time they are used saves buffer space. In some cases, display files have many formats but a job uses only one or two. Placing these formats into a separate display file (for example, based on application function) can reduce the PAG size which, in turn, reduces the number of disk I/O operations to read and write the PAG, and saves space while the PAG is in memory.

One conclusion you can draw is that insufficient main memory shared by multiple jobs can cause increased CPU and disk usage. This is non-productive and leads to diminished throughput and response time.

In some cases, you can avoid this scenario by not allowing pages to be stolen from other jobs. In the case of **batch jobs or non-interactive jobs**, you can run each job type in their own storage pool. Because storage management steals pages only from that job′s pool, it cannot get another job′s pages that are in another pool. If the pool still shows high page faults (especially database pages indicated by a high database fault rate in the pool), add enough memory to the pool to reduce the fault rates to an acceptable level.

In an **interactive** environment, you may be able to group jobs into separate storage pools based on unique application differences or separate remote from ″local″ attachment workstations using the application. Grouping remote jobs into a separate storage pool is recommended only when many of them are experiencing abnormal job termination. Typically, this is the situation for error-prone communication lines or many personal computers (PC) using workstation function and powering off the PC with several WSF sessions active. In large configurations, this can involve over 100 jobs abnormally ending at the same time.

The system attempts to control this situation by lowering the priority of the job termination processing, but it is still has a significant impact on the overall system performance.

For incoming program start requests (target job initiation) from communications devices, consider using pre-started jobs (Add Prestart Job Entry (ADDPJE) command). Properly designed pre-started jobs have already completed job initiation before the request is received and do not actually end when the conversation ends. Therefore, requests to use a pre-started job use much fewer resources than a request that requires full job initiation.

Similarly, rather than signing off (job termination) and signing on (job initiation), it is better to utilize DSCJOB. DSCJOB makes it appear as though a sign-off has taken place, but the job remains active. When the user returns and signs on (same user, same workstation), the ″suspended″ job is reconnected, thus eliminating the resource intensive termination and initiation. This may significantly improve performance.

11.8 Activity Level Performance Considerations

In analyzing activity level performance considerations, you need to understand that:

- You should use the recommended settings in *Performance Tools/400 Version 3*, SC41-4340 or Appendix A, “Guidelines for Interpreting Performance Data” on page 383.
- Changing the activity level value does not necessarily result in an immediate change in system performance.
- Too high a value can cause occasional performance degradation due to high paging, which can sometimes be severe.
- Too low a value can reduce interactive throughput and increase response times.
- Comparing Wait-to-Ineligible counts with Active-to-Wait counts gives a general view of how close the activity level setting is to optimum.

The user-specified storage pool activity level is one of the tuning adjustments available to keep the number of dispatch capable jobs within bounds. A proper activity level setting helps ensure that the load on other system resources (especially disk I/O and main storage paging) is kept within guidelines. This is true only in main storage constrained situations. With the improvement in the dynamic PURGE algorithm, activity level adjustments are not always as effective as they were before.

In a large storage pool, the initial activity level value should be calculated based on workload requirements (subsystem throughput and response time) rather than basing it on memory size.

Setting the activity level by dividing the interactive pool size by the estimated job size can result in an unrealistically high activity level value. For example, an AS/400 model 400 with an interactive pool of 200MB and an estimated job size of 1MB gives an activity level of 200. This is much higher than the guidelines. The effect of setting the activity too high is that almost every transaction arriving during a peak load situation starts processing as soon as it enters the system, which results in excessive disk and CPU demand, thereby delaying all transactions, even those already in progress.

To avoid pool overcommitment, it is better to set a reasonable activity level value and make new work wait a short time (perhaps 5 to 10% of the response time) before it starts. This allows the old work to finish before the new work starts. This strategy of **Favor Output Over Input** results in much better performance (more consistent and generally better response time).

Changing a pool′s activity level to adapt to minor changes in workload is not recommended. A constant need to adjust activity level says that something else is causing a problem and should be fixed (for example, resource intensive transactions, ad hoc work, high priority batch, and so on). The real constraint to activity level settings are high pool paging, re-paging (″thrashing″), and the resulting disk I/O usage. Once you find the proper value, set it and forget it.

11.9 Follow Specific Recommendations

Once you have determined the resource constraints (or confirmed Advisor recommendations and conclusions), you should make some changes. At this stage, it is recommended that you make changes that affect the resource (disk, CPU, long waits, and so on) you are dealing with.

Important

Make only one change at a time and measure the effects.

If your problem has been solved, keep the performance data for future requirements.

If problems remain after making a change, you need to go deeper into problem analysis and probably need to collect performance monitor trace data (STRPFRMON TRACE(*ALL)).

11.10 Main Memory and CPU Utilization - Tuning Roadmap

Balancing your main memory and CPU utilization is accomplished by allocating the available memory and setting the activity levels in the storage pools. Refer to the *OS/400 Work Management Version 3*, SC41-4306 for the initial setup of memory and activity level. The tables that contain information about fault rate guidelines are in Appendix A, “Guidelines for Interpreting Performance Data” on page 383.

Note: You have to repeat Step 4 through Step 7 for all of the other pools in your AS/400 system; Step 3 is for *MACHINE pool only. Follow the road map during periods of high systems activity because it is of little use to tune the system when there are few jobs in the system.

Interactive AS/400 Tuning Roadmap

1. Enter the WRKSYSSTS command.
2. Wait 2-3 minutes and press PF5 to refresh.
3. Does *MACHINE NDB faults meet the guidelines?
   a. Yes ... Press PF10 and go to Step 4.
   b. No .... Adjust QMCHPOOL:
      1) -100K if fault rate = 0
      2) +100K if fault rate > 5.0
      3) Press PF10 to reset and go to Step 2.
4. Does the DB faults + NDB faults for any pool meet guidelines?
   a. Yes ... Increase pool size by 100KB, press PF10 and repeat Step 4 (repeat until all pools are less than 20).
   b. No .... Go to Step 5.
5. Wait 2-5 minutes, press PF5. Is the Wait to Ineligible state = 0?
   a. Yes ... Reduce Activity level by 2, press PF10 to reset and repeat Step 5.
   b. No .... Go to Step 6.
6. Is the Active to Wait state 10 times the activity level?
   a. No ....System not heavily used or complex application mix go to Step 4.
   b. Yes ... Go to Step 7.
7. Is the sum of all fault rates for all pools within guidelines?
   a. No .... Go to Step 4.
   b. Yes ... Go to Step 8.
8. Activity levels and pool sizes probably OK. Continue monitoring WRKSYSSTS display regularly.

11.11 Operating Environment Tips

The following tips may prove useful when tuning your system:

IPL Your System: Occasional IPLs are no longer required on the AS/400 system for optimum operation. However, IPLing can assist in cleaning up some system areas and spooled areas. Permanent and temporary addresses are no longer regenerated because the number is now much greater using a 64-bit architecture. If the system runs out of addresses, the operating system must be reinstalled. Refer to the *AS/400 System Operation V3R6*, SC41-4203 for more information but try to IPL at least once a month.

Note: Temporary job structures are built during run time as needed according to system values QADLACTJ and QADLTOTJ described earlier in Section 11.3, “Review System Values” on page 184.

Storage Pool Tuning: Let the system do automatic tuning of shared pools by setting system value QPFRADJ to either 2 or 3. If you have user-defined storage pools, let the Performance Tools Advisor function evaluate storage pool information and follow its recommended changes.

If you do not use these system-provided tuning functions, you must monitor storage pool size and activity levels with the Work with System Status (WRKSYSSTS) command. Note that using database *sequential only* processing (discussed later in this section) may specify large block sizes that require increasing the pool size to reduce paging activity. Follow the guidelines in the *OS/400 Work Management Version 3*,SC41-4306, *Performance Tools/400 Version 3*, SC41-4340 or Appendix A, "Guidelines for Interpreting Performance Data" on page 383 of this publication.

If QPFRADJ is set to 0 during a period of poor performance, use the WRKSYSSTS command to see if any pool shows page faulting exceeding guidelines or more than 20% of the jobs in any pool go from wait to ineligible. Assuming this is the case, you should add memory to the pool or reduce the activity level. If you cannot move storage from one pool to another without causing more performance problems, you need more main storage.

Sometimes the system experiences a large number of page faults per second in the machine pool. When this lasts for several minutes, it is usually an indication that some kind of error logging is responsible, such as for remote communication lines. Here is a short summary of places to look for information:

- Messages to message queue QSYSOPR and log QHST
- Performance Tools report data: This includes:
  - Storage Pool Activity: Component report, Pool Interval report
  - Disk Utilization: System report, Component report, Resource Interval report
  - Communication Line Detail: Resource Interval report
  - Local Workstation Controller: Component report, Resource Interval report

  This data is collected only if the Performance Monitor is active.

  Any communication line error percentage greater than 5% should be resolved as soon as possible. You may need to vary off a problem communication line (including LAN lines) or a local workstation controller to eliminate the impact on other system work.
- Entries in the system error log: This log can be accessed through *option 2* from Start Service Tools (STRSST) or the Printer Error Log (PRTERRLOG) command. Some errors are self-explanatory and others require IBM service assistance.
- Work with System Activity (WRKSYSACT) command: If display updates continue to show task ERRLOG active, the system is reporting excessive error conditions.
- Some disk errors may only be indicated by an unusually high or low utilization of a disk, high disk IOP utilization, and high disk service time as shown in the Performance Tools reports.

Storage Pools for Page Isolation and Sharing: The single level storage model of the AS/400 system allows a single copy of code and records in a data space to be shared across all jobs in the system without having to make duplicate copies of the object as is commonly done in many other virtual memory systems.

Consider using the SETOBJACC command to load specific objects into a main storage pool or using expert cache support to have the system identify objects that should be loaded and held in the specific shared storage pool, especially for AS/400 models that are configured with large amounts of main memory.

The Set Object Access command temporarily changes the speed of access to an object by bringing the object into a main storage pool or purging it from all the main storage pools. If a separate pool is used and no other jobs are running in it, an object can be kept resident in main storage. This command can be used to cause a set of objects to be resident in a main storage pool (assuming enough main storage is available in the memory pool to hold all objects without excessive paging). For more information about using this command, refer to *OS/400 Control Language Reference Version 3*, SC41-4726, *OS/400 Work Management Version 3*, SC41-4306 and Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413.

Expert cache support requests the system to maintain reference pattern accesses for objects within a shared pool. If the system detects a high level of ″locality of reference″ (sequential storage locations) within a supported object, it ″pins″ the object in storage during this heavy usage, thus reducing physical disk I/O operations. When the system detects this locality of reference is no longer in effect, the object may be purged from main storage by other high locality of reference objects.

High CPU Utilization Jobs: Performance Tools Advisor and Display Performance Data functions and reports show high CPU utilization jobs if the Performance Monitor was active during the problem period.

The Work with Active Jobs (WRKACTJOB) command and Work with System Activity (WRKSYSACT) command can also identify active high CPU utilization jobs or tasks. If one of the jobs is running at interactive priority, the job may need to be held or have its priority lowered. In some extreme cases, the job may need to be terminated immediately through the ENDJOB command. If ENDJOB ... *IMMED does not complete, use the End Job Abnormal (ENDJOBABN) command to terminate the job.

Note: ENDJOBABN causes the next IPL to be considered as *abnormal* and should be used as a ″last resort″.

Monitoring Severe Messages: The system can detect certain conditions, such as exceeding the system ASP threshold, exceeding user ASP storage, hardware device starting to fail, and so on. Corresponding messages are sent to message QSYSMSG if it has been created by the user. Also in V3R6, ″Service Attributes″ (use DSPSRVA/CHGSRVA command) include the user profile class (multiple users) that receives these messages as well as the QSYSMSG message queue.

Monitor these messages as they indicate error conditions that can degrade performance, or even lead to abnormal system termination. These following messages should be monitored for their occurrence on a daily basis:

- CPI9490: Disk error. Disk starting to fail.
- CPI1138: User ASP overflowed. New data placed in system ASP.
- CPI0907: System auxiliary storage threshold exceeded.

  You can use WRKSYSSTS to observe the actual percentage of disk consumed. If the value reaches 99%, the system stops running.

Optimize Programs: The Create program commands and for ILE, the Create Module (CRTxxxMOD) commands provide optimization options that need to be understood from a high-level language (RPG, C, COBOL, CL) view and if ILE capabilities are being used. The optimization option may enable some form of reduction in the number of instructions actually run.

Since optimization can significantly lengthen the program compile time, it is recommended that optimization be selected only after the program is fully debugged.

An ILE CRTxxxMOD command provides three OPTIMIZE options (*NONE, *BASIC, and *FULL). In V3R6, ILE C has four levels of optimization. These are level 10, 20, 30, and 40. Levels 10, 20, and 30 are equivalent to *NONE, *BASIC, and *FULL respectively.

During ILE program development, we recommend using OPTIMIZE(*NONE) on the create command as some optimizations are always done, and this gives the shortest compile time.

In RPG and COBOL programs for both ILE and OPM environments, compute-intensive programs with significant looping, arithmetic operations, and string handling may experience up to 40% performance improvement compared to V3R1 for a system with a comparable RAMP-C or CPW rating. However, this was due to the architecture change, not due to optimization. Optimization gave nominal improvement with the best improvement in call intensive situations. The level of optimization is already quite good using *NONE, so applications should only be optimized toward the end of the development cycle.

In tests conducted by the ITSO with ILE C, using OPT(20) gave about 10% performance improvement at a cost of 1.7x longer compilation than OPT(10). Using OPT(30) gave about 45% performance improvement over OPT(10) at a cost of 4x longer compilation than OPT(10). OPT(40) gave about 5% performance improvement over OPT(30) with a similar compile time. OPT(40) gives greatest improvements when there are many procedure calls to small ″leaf″ routines.

Avoid Running Expensive System Services at High Priority: ″Expensive″ services include save/restore functions, large program compiles, long running display commands, queries, Open Query File command (OPNQRYF), file copy functions (CPYF), the Performance Tools Analyze Performance Data command (ANZPFRDTA), and so on. These should normally not be run at ″interactive priority″. Interactive priority normally defaults to 20. Batch type functions run at priorities equal to or higher than interactive functions, consume excessive CPU, and cause other jobs already in an activity level to have long waits before actually running.

These functions should be run in a batch subsystem by making use of the Submit Job (SBMJOB) command. The job should run with a priority lower than that used for interactive jobs. Another possibility is to run the function interactively but lower the priority of the current job (CHGJOB command) before running this function.

Note: Often these functions are run from the system console device. The system default priority for console device jobs is 10 and other interactive device priority defaults to 20.

Submit Queries to Batch: Query/400 supports submitting the defined query as a batch job. Enter the SBMJOB command and press PF4 to enter the RUNQRY command with all of the parameters. After that, press Enter twice to submit your query. This is recommended for long running queries to reduce their potential impact on interactive resources.

Use Mirroring: When mirroring is active, the system looks at each read request and chooses the unit that is least busy and has the shortest seek. Writes are executed twice (once to each arm). In general, mirroring adds a small amount of system overhead. In a primarily ″write-oriented″ environment, the overhead of mirroring needs to be considered. However, if reads constitute the dominant workload, mirroring can actually improve performance slightly as well as system integrity.

Mirroring provides the best disk protection and recovery options. However, RAID-5 protection may provide a significantly lower cost solution while still delivering excellent performance. Performance is reduced if a disk arm fails, and this should be taken into account when designing high availability high demand systems.

Minimize Authority Lookup Processing: For best security performance, use only public or owner authority for an object. This is because public and owner authority information is stored in the object. Private authority has the potential for significant performance degradation. Private authority may cause authority lookups within the process or job user profile to determine the level of authorization. These lookups are system index operations and are accounted for in the Performance Tools reports. See the CPU impact of authority lookups in Appendix B, “Program Exceptions” on page 387.

Authority lookup processing remains a potentially high CPU usage function when private authorities are used for frequently accessed objects. Some general guidelines when using an AS/400 system private authority support include:

- Use only public or owner authority on objects for best performance.
- Use private authority only on objects that should be secured. Do not generally use private authorities on IBM-supplied objects such as message files and programs. These objects are shipped with adequate protection and it is important that those message files and programs that are used frequently involve minimal security processing.
- If you use private authority, use a private authority that is greater than ″public authority″ for that object. If the private authority is less than public authority, the system takes more CPU to verify adequate authority.

  With private authority, the system has to process both the object and the accessing user′s profile. If that profile has many ″private authorities″, processing the user profile can become excessive when the private authority is less than public authority.
- Using group profiles or authorization lists provide ease of maintenance and record keeping. They do not improve authority lookup performance. If an authority list is used to secure an object and private authorities exist for that object, the number of authority lookups can increase as discussed previously for private authority less than public authority. Group profiles can cause an increase in authority lookup because the process (job) user profile is examined before the group profile.

> Version 3 Release 6 supports **supplemental groups authority** and **primary group authority** that require additional authority lookup performance consideration.
>
> Supplemental groups enable a user to be part of more than one group profile. This is supported through the Create User Profile (CRTUSRPRF) command SUPGRPPFR parameter. This is an important ease-of-use enhancement but can lead to significant performance degradation if many supplemental group profiles are specified for an object created by the user profile with these supplemental groups.
>
> Primary group authority identifies the primary group profile and that profile′s authority level for an object. This authority information is stored with the object and can provide a *fast path* for private authority lookup for group profiles.
>
> Both the *AS/400 Security Reference - V3R6*, SC41-4302, and the redbook *An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, Communications, and PC Connectivity*, GG24-4200, provide more details and performance considerations for using supplemental group and primary group profile authorization.

If Authority lookups are suspected of causing excessive CPU usage, the IBMLIB AUTHTRC command and AUTHPRT command can be used to find which objects and jobs are causing this problem. See Appendix C, “IBM Internal Use Only Tools/Documents” on page 393 for more details.

Adjust Interactive Job Time Slice: The default class description used for interactive jobs is QINTER and its TIMESLICE value is 2000 milliseconds. For almost all interactive workloads, two seconds is acceptable because interactive transactions *should not* last longer than two seconds. However, there are cases during heavy interactive workload, where lowering the time slice to .5 seconds may improve performance.

Using Performance Tools/400 helps you to determine the correct value of TIMESLICE on your system. From the System report, you get average CPU seconds per interactive job type; multiply the number by three and set the result to the TIMESLICE value.

Consider setting interactive job TIMESLICE to .5 or less seconds on the faster AS/400 processors.

Use DEVRCYACN or Program for Workstation Errors: Programs that communicate with the requester display device may cause high CPU utilization when they fail to detect an error on the requester device. These programs can loop, continuing to write and read to the device even though no useful function is being performed. If a large number of these jobs are looping at the same time (for example, when an entire remote line fails), impact on the entire system can be severe. The system detects this looping after a period of time and automatically terminates the job at a reduced priority. However, programming for workstation errors and using job attribute DEVRCYACN can further reduce CPU impact.

The most efficient technique is to use a combination of the job attribute DEVRCYACN(*DSCMSG) and add user program code to check for workstation (RPG) or transaction (COBOL) file major/minor return code 83E1. This technique suspends (disconnects) the job when the error is indicated to the job. When the line/controller/device is recovered, the system puts up the sign-on display. If the operator signs on with the same user ID as the suspended job and requests reconnection to the suspended job, the program is notified with 83E1. 83E1 indicates the display contents have been destroyed and the program should ″re-start″ the ″in-process″ transaction.

This combination requires the most programming effort but significantly minimizes CPU impact on all system activity when a workstation error affects many workstation jobs.

DEVRCYACN values *DSCENDRQS, *ENDJOB, and *ENDJOBNOLIST can be used without user coding for major/minor return codes as the next best level of reduced CPU impact.

DEVRCYACN(*MSG) is the default and returns control to the affected job which can loop with failing workstation I/O operations if the error message or major/minor return code is ignored. However, data management automatically detects five successive I/O failures and automatically terminates the job. When this job termination is performed, as well as DVRCYACN(*ENDJOB or *ENDJOBNOLIST), the job priority is dropped and time slice changed to 100 milliseconds to minimize the impact of a large number of jobs terminating at the same time.

Note: In V3R7 the default value is DEVRCYACN(*DSCMSG) which disconnects the job. When signing on, an error message is sent to the user′s application program.

System value QDEVRCYACN can be set to a value that is used for all requester workstation jobs from non-programmable workstations and Personal Computer Work Station Function (WSF) sessions.

For Display Station Pass-Through jobs, DEVRCYACN is ignored and the job is immediately terminated by the system.

Note: In a completely uncontrolled environment, the suspended jobs resulting from *DSCMSG may keep files open and retain record locks that may prohibit other jobs from running. System value QDSCJOBITV can be used to end the disconnected jobs after a period of time. The WRKCFGSTS command and WRKSYSACT command do not show disconnected jobs. The WRKACTJOB command shows the disconnected jobs if F14 is used.

Display Station Pass-Through: In some distributed environments, a decision has to be made as to whether to use Display Station Pass-Through or Distributed Data Management. In cases where only a few records need to be accessed on the remote system, DDM is quite acceptable. If a large number of disk records need to be accessed on the remote system, Display Station Pass-Through is recommended.

In special situations, APPC programming within the workstation job to access the remote data may be worth the programming investment.

Using the Transfer Pass-Through (TFRPASTHR) command within a program can make return to the local system transparent to the local workstation operator and make that return much faster.

Use CL MONMSG to handle the errors rather than having the system terminate a job (that is, at wait time out) and forcing another sign-on.

Consider Use of Disconnect Job: In some environments, there is a need for several sign-on and sign-off sequences during a normal day. Sign-on and sign-off use large amounts of CPU and disk resource. Consider using the DSCJOB command function. This suspends the job and shows the sign-on prompt display again. The next user on that workstation may sign on and reconnect to the suspended job (same user ID, same password) or begin a new job. Reconnecting to the suspended job minimizes system overhead.

See “Use DEVRCYACN or Program for Workstation Errors” on page 203 for information on automatically terminating a disconnected job through the QDSCJOBITV system value.

Restrict Generic Searches: It is not unusual to have a generic search function as part of an application, for example, a customer name search. Used correctly, such searches result in accessing a limited number of records. A lazy operator can generate a large number of I/Os by habitually looking for all of the J′s or S′s rather than JOHN* or SMIT* .

If you must use generic searches, use a ″starts with″ (for example, ABCdefgh...) technique, rather than an ″is contained in″ (for example, abcDEFgh...) technique.

Use Prestart Jobs: Prestart jobs can increase the responsiveness of the system to incoming program start requests (incoming ″evoke″ or ″attach″) received from communications devices. A prestart job permits the system to have already completed job initiation overhead and the initial application program to have completed its startup overhead processing (such as file open) *before* the evoke is received.

Prestart job support is implemented through entries in the subsystem description through the Add Prestart Job Entry command. For programming considerations, refer to the *OS/400 APPC Programmings*, SC41-3443, and for SAA CPI-C, the *Common Programming Interface - Communications Reference*, SC26-4399.

See also Section 12.10.3, “Using Pre-Started Job Support” on page 290 for additional considerations.

Use 5250-Type Workstations Rather Than 3270-Type Devices: The 3270 device support is for the following devices:

- 3174 establishment controller
- 3274 control unit
- 3270 display stations and printers
- All emulators conforming to 3274 Control Unit Model 31C
- 3287 printer
- Personal computers using 3270 emulation

3270 displays require the system to translate between 5250 data streams and the data stream understood by the device. This translation time is rather small (for example, 30 milliseconds). However, there is significant buffer storage required for 3270-5250 data translation. As the number of active 3270 displays increases, additional buffer storage requirements may cause significant performance degradation on limited main memory machines.

There are certain 5250-based applications that result in excessively long 3270 data streams sent from the AS/400 system under 3270 Remote Attachment support. Default processing on the AS/400 system is used to compress the application′s 5250 data stream before translating to the 3270 data stream. This reduces the data stream length sent to the 3270 display device at a small cost of CPU utilization overhead. The amount of data stream length reduction varies, based on the way different DDS functions are used.

In some AS/400 run-time environments, the additional CPU required to compress the 5250 data stream cannot be tolerated. 5250 data stream optimization support is used by default for 3270 translated data when the system communicates with 3270-type displays using the following communications support:

- 3270 remote attach
- Distributed Host Command Facility (DHCF)
- Network Routing Facility (NRF)
- SNA Primary LU2 Support (SPLS)
- TELNET

Controlling 5250 data stream optimization support is included in V3R6. A data area of the same name as the control unit description is required. This data area must be defined as a 15-character data area in library QSYS. To turn off optimization, the data area must contain ″OPTDTASTR(*NO);″ to turn on optimization after it has been turned off, the data area must contain ″OPDTASTRM(*YES)″.

Create Control Unit (CRTCTLxxx) or Change Control Unit (CHGCTLxxx) looks for this data area and sets up appropriate internal control block information. If the data area does not exist, optimization is in effect. There is no way to display the control unit to determine its current data stream optimization value.

The following example is a data area for control unit CTL01 with optimization turned off.

```
CRTDTAARA DTAARA(QSYS/CTL01) TYPE(*CHAR) LEN(15)
VALUE(′OPTDTASTR(*NO)′)
TEXT(′THIS IS THE 5250 OPTIMIZATION VALUE FOR CTL01′)
```

You must have at least *USE authority to the data area object to use the create or change controller command. The data area must be created with authority *USE or higher.

For more information on using 5250 Data Stream Optimization Support, see Chapter 11, ″Using 3270 Device Support″ in *Remote Work Station Support V3R1*, SC41-3402.

Avoid Logging CL Commands and Outputting Job Logs: In a production environment, logging control language (CL) commands during application activity and generating job logs at job termination can place excessive demands on system resources when done for many jobs at the same time. These functions are controlled by job description parameters.

Spooled File Overhead: The following recommendations can improve spooled file performance:

- Keep the number of spooled files remaining on the system to a minimum.

  System performance can be degraded when thousands of print files remain on the system. See the index entry *system value* for information on automated spooled file cleanup (QRCLSPLSTG). You may also use the GO ASSIST menu to schedule cleaning up job log and problem dump spooled files.

- If AFP support (now known as PSF/400) is not required, create the printer device description as AFP(*NO). This eliminates unnecessary print driver job overhead.
- If spooled file data is intended for an IPDS printer, specify DEVTYPE(*IPDS) on the CRTPRTF, CHGPRTF, or OVRPRTF command to avoid the unnecessary system processing of transforming the SCS printer data stream (default) to IPDS at the time of print.
- When there is heavy use of AFP printing on a system, we recommend to set pool storage used by subsystem QSPL to approximately 8MB.
- When using AFP overlay support, group print files use the same overlay in sequence for printing:

  This enables V3R6 printer data management to send the overlay to the printer for the first file. The overlay is not sent for the succeeding print files using the same overlay. This improves overlay performance compared to previous releases.
- When using ASCII printers, consider not using *host print transform* support:

  ASCII attached printers emulate specifically supported printers. These ASCII printers provide the emulation support without requiring AS/400 CPU utilization. Use AS/400 host print transform only when AS/400 CPU impact is acceptable and AS/400 ASCII printer emulation support provides better emulation.
- When using the SNDNETSPLF command, consider using value *ALLDATA instead of *RCDDATA for the parameter DTAFMT. The reason for this is that with *ALLDATA, the OS/400 does only a fraction of the amount of data conversion compared with *RCDDATA.

  Be careful of the value of the DTAFMT parameter when sending files to computers on previous releases of OS/400 since all of the values may not be supported.

Save and Restore: For best overall system performance, always understand what other system work is active during saving or restoring, the amount of storage pool space available, and the speed capabilities of the tape or disk device being used.

When planning for the save function, ensure a reasonable amount of main storage can be dedicated to the save function.

You must consider the following when setting up or modifying a ″save environment″.

1. Tape device and controller speed and data compression capabilities:

   Hardware data compression provides significantly faster throughput rates and less CPU utilization versus software compression. When other variables are identical, the 3590 provides the fastest rates.
2. Type of object being saved:

   A single large object can be saved faster than a set of objects when generally the total bytes of data being saved are the same.
3. CPU processing speed:

   Though much of the save/restore process involves disk I/Os, increasing CPU processing speed does improve expected rates.

4. System disk and tape device configuration:

   In most hardware configurations, the attachment configuration of system disk and tape device enables maximum throughput.

   However, there are two configurations that have shown degraded save or restore performance:

   - 3490/3590 tape device and system disks attached over the same I/O bus:

     Maximum 3490 rates cannot be achieved if the 3490/3590 and system disks are attached over the same I/O bus. To achieve maximum 3490/3590 throughput, you must place the disks and the 3490/3590 device on a separate bus.

   - 6390 internal tape device and system disks attached to same Multi-Function I/O Processor (MFIOP):

     Restoring from the 6390 (8mm) tape is 10% to 20% slower when compared to its equivalent external 8mm tape device (the 7208-12 attached to the 2621 IOP). To achieve equivalent performance for the 6390 tape device, the disks must be moved from the MFIOP. If this is not possible and 8mm tape support is required, you must use the 2621 IOP and 7208-12 tape configuration for maximum performance.

5. Available storage pool memory:

   A save/restore session can take up to one-half the available memory within a storage pool. In general, a save/restore session with up to 2MB of storage available gets at least a moderate level of throughput. At 12MB of available storage, a single save and restore session achieves close to its maximum throughput. If multiple save and restore sessions are to be concurrently active, allow approximately 12MB for each concurrent save and restore session.

   If a nightly backup is planned, when no other significant work is being done on the system, give all of the available storage to the save job.

6. Save and restore job priority:

   Setting run priority of a save and restore job to ″high″ gives more processor power to this job (or jobs) and achieves improved throughput.

   However, running any ″batch type job,″ such as a save or restore, at a priority higher than an interactive job degrades the interactive job′s performance. As previously discussed, scheduling save and restore function is critical to realistic customer satisfaction.

7. Save/Restore in RAID-5 or mirroring:

   RAID-5 and mirroring all have an impact on the restore function compared to ″no DASD protection″ environments. This should be expected and included in any backup and recovery plans.

   Mirroring provides the most DASD protection and restore is in most cases faster than RAID-5 support when other variables remain constant, such as the tape device and objects restored. In a mirrored-environment laboratory test, results showed save times with mirroring can range between three percent faster to five percent slower when compared to no DASD protection.

System tuning should be conducted to achieve the maximum performance that can be achieved in the ″normal backup″ mode of operation.

See Section 13.5.5, “Save/Restore—Planning” on page 347 for save restore performance considerations for data stored in the new Integrated File System.

The *AS/400 Performance Capabilities Reference for PowerPC Technology*, ZC41-0607 contains test results for many save and restore configurations.

11.12 Communication Performance Considerations

From a performance perspective, the basic communications considerations are:

- Maximize line speed.
- Minimize the amount of data sent and received.
- Minimize the number of line turnarounds per unit of work.
- Minimize unproductive line time, such as waiting for inactive control units to respond or time spent re-transmitting data due to line errors.
- Set realistic expectations based on the line speed, link level protocol (SDLC, Asynchronous, and so on) and the data flow protocol (SNA, TCP) as implemented on the AS/400 system.

  For V3R6, the AS/400 system supports:

  – LAN token-ring (TRLAN)
  – LAN Ethernet (ETHLAN)
  – LAN Fiber Distributed Data Interface (FDDI)
  – LAN Shielded Distributed Data Interface (SDDI)
  – Synchronous Data Link Communications (SDLC)
  – Binary Synchronous Communications (BSC)
  – Asynchronous (ASYNC)
  – Integrated Digital Loop Carrier (IDLC)
  – X.25 packet switching (X.25)
  – Integrated Services Digital Network (ISDN)
  – Frame Relay (FR)
  – Wireless LAN
  – SNA protocols
  – TCP/IP protocols
  – Open Systems Interconnect (OSI) protocols

Note: This redbook provides some performance information on most of the protocols but provides details only on SDLC, LAN token-ring, SNA 5250, SNA APPC, and TCP/IP link level and application level protocols.

This information includes SNA protocol and token-ring data buffering and parameters, and application design and coding techniques that can minimize the amount of unnecessary data transmitted and the frequency of line turnarounds that can lead to improved communication performance.

When examining communication performance, first analyze the line and control unit line protocol parameters, the higher level protocol parameters such as SNA Maximum RU Length and Pacing values, and finally the application design. Application design includes the number of database accesses, any waiting for database record locks, data area or message queue access delays, communication record length and blocking designs, and the number of application-to-application I/O operations necessary to complete a ″transaction″ or other performance metric.

Note: Sending data in the largest possible line protocol frame size (or packet size) is the single most critical performance consideration as this minimizes line traffic overhead and CPU utilization within the AS/400 system.

Review the following V3R1/V3R6 manuals for additional protocol-specific information:

- *OS/400 Communications Configuration V3R1*, SC41-3401
- *OS/400 Remote Work Station Support V3R1*, SC41-3402
- *OS/400 ISDN Support V3R6*, SC41-4403
- *OS/400 Local Area Network Support V3R1*, SC41-3404
- *OS/400 X.25 Network Support V3R6*, SC41-4405
- *OS/400 Communications Management V3R1*, SC41-3406
- *OS/400 TCP/IP Configuration and Reference Version 3*, SC41-3420
- *OS/400 Sockets Programming V3R6*, SC41-3422
- *OS/400 ICF Programming V3R1*, SC41-3442
- *CPI Communications Support*, SC26-4399
- *Performance Capabilities Reference for PowerPC Technology* , ZC41-0607

The OS/400 Performance Monitor collects communication line and IOP data transmission, error recovery and utilization statistics, and records the data in the QAPMxxxx family of database files. In OS/400 V3R6, performance data can be collected for SNA, SNADS, APPN, and station-level database files and 5494 remote control unit response time information.

Note that STRPFRMON RRSPTIME(*SYS or response time range values) must be specified to collect 5494 response time data.

The performance monitor also collects and records data from the Integrated PC File Server (FSIOP).

Table 20 on page 211 lists the communications-related QAPMxxxx database files and the type of information recorded.

Table 20. Performance Monitor Communication Data Files

Performance Monitor File	**File Content**
QAPMAPPN	APPN related performance data
QAPMASYN	ASYNC protocol performance data
QAPMBSC	BISYNC performance data
QAPMCIOP	Communications processor performance data
QAPMDDI	FDDI/SDDI Performance Data
QAPMECL	TRLAN performance data
QAPMETH	Ethernet LAN Performance data
QAPMFRLY	Frame Relay performance data
QAPMHDLC	HDLC (SDLC) performance data
QAPMIDLC	IDLC performance data
QAPMIOPD	FSIOP performance data
QAPMJOBS	Job related performance data (contains line and control unit information only for interactive jobs and DDM - no APPC jobs)
QAPMLAPD	LAPD performance data
QAPMLIOP	Local WS Processor performance data
QAPMMIOP	Communications processor performance data
QAPMRESP	Local workstation response times
QAPMRWS	Remote workstation (5494 controller only) response times
QAPMSAP	TRLAN / EtherLAN SAP Performance Data
QAPMSNA	SNA related performance data
QAPMSNADS	SNA/DS performance data
QAPMSTND	FDDI/SDDI Station Performance Data
QAPMSTNE	Ethernet Station Performance Data
QAPMSTNL	TRLAN Station Performance Data
QAPMSTNY	Frame Relay Station Performance Data
QAPMTSK	LIC task related performance data (contains line and control unit (station) name)
QAPMX25	X.25 performance data

Performance Tools/400 produces reports based on the communication IOP (including Integrated PC File Server) and line and 5494 remote controller response time data. No reports are produced for the SNA, SNADS, APPN, Station Level, and Integrated PC File Server data.

Performance Tools/400 Version 3, SC41-4340, contains a chapter on SNA, SNADS, and APPN counters that are collected by the OS/400 Performance Monitor, but **not included in any Performance Tools/400 reports.** You need to study the field definitions and determine what queries to define and run to obtain statistics. There are no guideline values given for these fields so they are useful only if you save the output over time and detect any trends in increasing values.

A communications performance residency was completed for V3R1 level support. A redbook was produced from this residency that documents sample queries and gives more communications performance information. Refer to *AS/400 Communication Performance Investigation*, SC41-4669.

Both *OS/400 Work Management Version 3*, SC41-4306 and *Performance Tools/400 Version 3*, SC41-4340 provide a list of all communications related database files and brief field descriptions for the data collected by the OS/400 Performance Monitor.

Communication line type protocols have parameters that control frame sizes, the frequency of frame acknowledgements, and timeout and retry values. In general, the more data that can be sent without turning the line around for acknowledgements and less re-transmissions based on communication error recovery, the better performance is realized.

11.12.1 Using SST to Perform a Line Trace

Higher level protocols, such as provided under TCP/IP and SNA, that run over the lower level communication type protocols can also impact performance.

The System Service Tools (STRSST) command provides a line protocol trace that can also subset its output to show formatted SNA-level data. In some cases, only this trace and a knowledge of the protocols involved can verify what is actually happening on the line. Consider two examples (actual frame size used and SNA ″definite response mode″ (discussed later in this redbook)).

A line-level protocol trace is required to determine the actual frame sizes used and line turnaround frequency. An SNA-level trace listing can be used to observe the use of ″definite response mode″ or ″exception response mode″ (less line turn-arounds).

The SRTSST line trace with an SNA format option for printing omits frames that do not contain SNA-formatted data. On an SNA frame, the acronyms DR1 and ERI mean exception response mode is being used. If ERI is not shown, definite response is being used. Note that for certain SNA data exchanges, such as the SNA BIND command, definite response mode is required. However, for a frame that contains normal application data, ERI should be seen.

A protocol level line trace, such as SDLC, includes, but does not differentiate SNA data. With this level of output, time stamps for Send and Receive frames are listed. These time stamps (in 100 millisecond increments) can be used to tell how long it took for application and system processing to respond to incoming data. The time stamps are not correlated with the time-of-day clock in the system, but can be used to determine the time differences between various data transmissions and receptions.

The *Data Collection Guide*, SC21-8253, discusses how to use the system service tools, including how to collect communication trace data.

11.12.2 Communication Line Speed Considerations

The maximum speed capability of any communication line only becomes important when the communications line becomes a bottleneck. For interactive environments, line utilization of up to 50% can deliver acceptable response time, although performance begins degrading above 40%. For batch environments, line utilization approaching 100% is desirable.

Remember that line utilization is defined as the actual measured throughput in bits per second (bps) divided by the rated line speed. Often (although incorrectly), people translate bits per second to characters (bytes) because it is easier to relate characters to application data. The number of characters transmitted includes data resent because of line errors, communication protocol overhead data, actual application data, and device-specific overhead data such as field definition characters for display devices. The OS/400 Performance Monitor collects communication data and the Performance Tools System report and Resource Interval report show line utilization and error summary information.

Multiple jobs exchanging data concurrently over the same line may be necessary to maximize utilization of the communications line. Multiple jobs on a single system are necessary to make use of the system and adapter speed capability on the LAN. At LAN speeds, the application code, system communication code, and the I/O processor (IOP) itself may become throttles limiting the maximum transmission rate possible with a single system.

The newer communications IOPs (2617, 2618, 2619, 2623, 2665, 2666, 2668, and 6516) are necessary to make use of the ″IOP performance assistance″ functions available in V3R6 to improve APPC and TCP/IP performance. Many of the communications management functions, such as assembling and disassembling SNA RUs within frames, are performed by these IOPs and thus reducing the resource that otherwise must be handled by the AS/400 CPU.

11.12.3 Line and Control Unit Error Recovery Parameters

The various protocols and controller types have error retry and timeout values. Lines also have threshold values that can be used to indicate recoverable errors over a period of time when the retries specified in the configuration object have not been exceeded in consecutive retries.

When a consecutive series of retries has exceeded the retry count, a message is sent to any affected jobs and message queue QSYSOPR and log QHST. This is considered ″first-level error recovery″. While the affected jobs may or may not be programmed to recover from the indicated error, the system *attempts* to recover the line or control unit to a ″varied-on″ state. This is considered ″second-level recovery″. The system uses either the system value QCMNRCYLMT or the object description value for the CMNRCYLMT parameter. Normally, the system′s attempt at recovery utilizes minimal system resources. However, if the values of this second-level ″recovery limit″ are mis-configured, there is a potential that this second level of recovery consumes significantly more resources than normal.

For example, a CMNRCYLMT value of (2 5) is normal and means ″attempts to recover the first-level (line or control unit retry value exceeded) failure two more times within five minutes″. If this second-level of recovery fails twice within five minutes, the system ceases recovery attempts and places the object in a ″recovery pending″ status that requires operator action.

If CMNRCYLMT is changed to something such as (n 0), where ″n″ is any value, the system attempts recovery ″forever″ because zero means there is no time limit for the second-level recovery. This causes serious system performance degradation and can only be stopped by varying the configuration object off.

11.12.4 LAN Line and Control Unit Parameters

While application design, job run priority and SNA MAXLENRU (maximum SNA RU length, and SNA Pacing values impact interactive and batch performance, LAN protocol parameters also play an important part in performance on the LAN and should be analyzed first.

Application design and SNA values are discussed elsewhere in this redbook. This section discusses LAN-specific information, highlighting LAN line, control unit, and I/O Processor (IOP) information.

The following paragraphs present LAN-specific guidelines that highlight the importance of LAN frame size (MAXFRAME), maximum number of frames that can be *received* before sending an acknowledgement (LANACKFRQ), and maximum number of frames that can be *sent* before waiting for an acknowledgement (LANMAXOUT). The line description specifies maximum frame size supported by the LAN line. The control unit description may specify a frame size less than or equal to the line frame size and LANACKFRQ and LANMAXOUT values.

- Use the maximum frame size (MAXFRAME on the line description) wherever possible. Note that with token-ring LAN protocols, the first active station on the line sets the LAN line speed for subsequent attachments. LAN bridges may limit the maximum frame size and line speed. Understand the capabilities of the bridge before setting ″remote LAN line″ performance expectations.

  On the AS/400 system, the following maximum frame sizes are possible:

  - 16393 bytes for the 4/16 Mbps token-ring adapter feature (IOP) 2626 or 2619 when running 16 Mbps:

    Note that the 6516 IOP (token-ring or Ethernet) has the same performance rating as the 2619 for token-ring protocol and the 2617 IOP for Ethernet LAN. The 6516 is the Integrated PC File Server (FSIOP) used with LAN Server/400.

  - 1496 bytes when using Ethernet LAN adapter 2617 at 10 Mbps

  - 4444 bytes when using FDDI LAN adapter 2618 at 100 Mbps

- Control unit MAXFRAME (less than or equal to the line MAXFRAME values) and LANACKFRQ and LANMAXOUT are the primary ″performance parameters″ when the CPU model, application design, and SNA Maximum Length RU value remain constant:

  - *Never* allow LANACKFRQ on one system to have a greater value than LANMAXOUT on the other system.

  - In general, LANACKFRQ=1 and LANMAXOUT=2 (the parameter defaults) offer the best performance for interactive LAN environments and adequate performance for large file transfer environments.

  - For dedicated large file transfer environment, changing the LANMAXOUT value to 6 may provide a significant performance increase.

    In some cases, you may need to experiment with different values as discussed here:

    - When using the newer LAN IOPs (that is, 2617, 2618, 2619, and 6516) test with MAXLANOUT set to 6. See if performance is better than LANMAXOUT(2). If performance has improved over LANMAXOUT(2), test with LANMAXOUT(4). If performance is not significantly better than 2, use LANMAXOUT(2), which is the default.

    - Lab tests show little performance improvement occurs with LANMAXOUT set to 8 or higher.

    - When communicating with high performance PC′s, increasing LANMAXOUT above 2 may yield improved performance, but keep

MAXACKFRQ at 1 (which is the default). For personal computers with lower performance, use LANACKFRQ(1) and LANMAXOUT(2) as this reduces the incidence of buffer overruns.

Note: In mixed environments, LANACKFRQ(1) favors the interactive data exchange at the expense of batch transfers. LANACKFRQ can be increased to improve batch transfers, but interactive data exchange is degraded.

General considerations for the LAN IOPs include:

- The 6516 Integrated PC File Server I/O Processor (FSIOP) provides the fastest LAN performance with the 2618 FDDI IOP a close second. The limited maximum frame size of the 2618 FDDI IOP probably accounts for the slightly slower performance.
- The 2619 TRLAN IOP provides high performance at a lower cost than the 6516 Integrated PC File Server that also contains file serving software.

  The 2617 Ethernet IOP is slightly faster and has marginally greater throughput than the Ethernet adapter on the 6516.
- For interactive environments, keeping IOP utilization at 60% or less is recommended. Exceeding this threshold in a large file transfer environment may still offer acceptable performance.

  Use the Performance Tools/400 System Report and Resource Interval Report to observe line and IOP utilizations. The Advisor function also highlights possible utilization threshold problems.
- In heavy LAN data transfer environments, always consider spreading workload across multiple IOPs.

Table 21 on page 216 shows the highest achievable LAN data transfer rates for a single program, one-way transfer for most of the supported LAN IOPs. Data was sent from an AS/400 500-2141 to an AS/400 500-2142. IOP feature numbers are listed in parentheses. Each benchmark involved 16 384 sends of 2048 bytes each (that is, 32MB).

Table 21. AS/400 LAN - Maximum Throughput Capabilities

PROTOCOL (IOP)	**Line Speed (Mbps)**	**Frame Size (bytes)**	**Data Rate (kbps)**	**CPU Util Source/Target (%)**
TRLAN (2619)	16	265	1743	6.8/9.2
TRLAN (2619)	16	1024	5065	20.0/16.1
TRLAN (2619)	16	1994	7457	29.4/20.7
TRLAN (2619)	16	4060	9256	36.6/25.5
TRLAN (2619)	16	8156	10737	42.8/29.5
TRLAN (2619)	16	16393	11671	46.8/32.5
Ethernet (2617)	10	1493	7895	31.2/24.2
FDDI (2618)	100	4105	15790	63.0/42.8

Note:

This table is a subset of a table from the *Performance Capabilities Reference* manual. The rates shown are the best rates achievable and include no database I/O in the application. The largest supported frame size and best values for SNA MAXLENRU and PACING and Frame Relay LANACKFRQ and LANMAXOUT were used.

Customer applications may realize lower rates than those shown as they typically perform many database I/O operations and use significantly shorter user program record lengths.

11.12.5 SDLC Line and Control Unit Parameters

SDLC applications have similar frame size and frame acknowledgement, SNA data flow protocol, and application design considerations as discussed in Section 11.12.4, “LAN Line and Control Unit Parameters” on page 213.

The AS/400 system provides SDLC frame size control through MAXFRAME, which defines the number of frames that can be sent before an acknowledgement (MAXOUT), and the number of ″MAXOUT sets of frames″ that can be exchanged with the same control unit before communicating with the next control unit on a multipoint line (POLLLMT and OUTLIMIT). In cases where two systems are communicating with each other, setting these values can be quite flexible. In cases where the AS/400 system is communicating with a remote workstation controller, such as the 5394 or 5494, the controller supports smaller maximum values than the AS/400 system.

There are also SDLC polling timer and retry values that can control the impact of a non-responding controller on other active control units on the same line. Included among these are the CNNPOLLTMR, CNNPOLLRTY, and NDMPOLLTMR parameters that apply to the system acting as the **primary** (polling) station on the multipoint line.

The *OS/400 Communications Management V3R1*, SC41-3406, has a thorough explanation of how these parameters interact and provides some examples. In general, consider the following information:

- Use the configuration defaults for most parameters until the remote communications environment is understood. The line type protocols are one of the ways to control the frequency and amount of data exchanged with each control unit. SNA parameters such as Request/Response Unit size (MAXLENRU) and PACING, and user control of line turnarounds through device file parameters (such as RSTDSP and DFRWRT) and program output operations can sometimes have a greater impact than the configuration parameters.

- For SDLC multipoint lines at speeds of 9600 bps or higher where it is typical for more than five control units to be powered off at any one time, you must consider either varying on the control unit only when it is likely to be powered on or specify explicit values for the line description CNNPOLLTMR and the control unit description NDMPOLLTMR parameters. This action can minimize the impact on active controllers when the system polls these *inactive* control units.

  For example, if the default of CNNPOLLTMR (30 - 3 seconds) is used every time a powered off control unit (VARY ON PENDING status) is polled, no line activity occurs for three seconds. Assuming seven of these powered off control units are polled in succession, there could be at least 21 seconds where no work can be done by the active stations on that line. By specifying CNNPOLLTMR(5), the system waits a maximum of one-half second for each control unit or experiences 3.5 seconds of inactivity for seven of these powered off control units.

  For control units that are active for only short periods of time throughout the day, consider using NDMPOLLTMR (900 - 90 seconds). This causes the system to poll this control unit approximately every 90 seconds. Using 90 seconds does have a disadvantage for the control unit when it is powered on. This is because it may be 90 seconds or more between the time the control unit is powered on and the system sign-on prompt is displayed on an attached workstation.

  As an alternative to using the configuration parameters to control when the system periodically polls a powered off controller, the customer may be able to automate a procedure that varies on the control unit when it is most likely to be ready for activity. This technique works most satisfactorily when a control unit becomes active based on a ″start of work day″ schedule across time zones.

  If you think polling timeouts for powered off controllers may be a problem, run the Performance Monitor and use the Performance Tools Advisor or Resource Interval report to analyze time spent polling without a response. Refer to the column ″Pct Poll Retry Time″ on the Resource Interval report. The percentage listed indicates the portion of the Performance Monitor sampling interval consumed by waiting for non-responding control units.

- Use the maximum frame size (MAXFRAME on the line description and control unit) as supported by the specific control unit. If a line is subject to frequent error conditions, you may need to use smaller frame sizes while pursuing a resolution to the error conditions.

  Use the Start Service Tools (STRSST) command to trace the communication line and verify the line protocol (and SNA) parameters *actually used*.

- The most common implementation for receiving the maximum number of SDLC frames before requiring an ″acknowledgement″ (line turnaround) is seven. Normally, once the primary station receives a satisfactory acknowledgement, the next control unit is either sent pending data or polled for input. This is the default for AS/400 configuration objects and is satisfactory in over 90% of all configurations.

  In some application environments, seven frames may not be sufficient for a control unit to receive all output data currently available from the host

AS/400 system. In that case, consider specifically setting the OUTLIMIT on the appropriate control unit to 1 or 2. OUTLIMIT(0) is the default and means up to seven frames are sent and the next control unit is communicated with.

The typical scenario for using 1 or 2 for OUTLIMIT is when seven frames to a control unit results in partial displays on attached displays. This is commonly upsetting to the workstation operator if there are many other active controllers on the same line. With OUTLIMIT(0) in effect, the system must communicate with all of those controllers before resuming transmission to the control unit with partial displays. OUTLIMIT(1) or OUTLIMIT(2) enables the system to send up to 14 or 21 frames respectively to that control unit before going on to the next control unit. If the system has fewer frames to send than OUTLIMIT supports, the next control unit is contacted just as before.

Changing the default of OUTLIMIT for a control unit should be done only after the applications used with that control unit are well understood as misusing OUTLIMIT permits a specific control unit to monopolize the available line capacity.

11.12.6 General Remote Communications Considerations

While in-depth coverage of all AS/400 supported remote line protocols is beyond the scope of this redbook, some summary level information is provided here, along with a sample of lab performance test results of a batch application over the various supported line protocols.

Note that the term now used to address all ″remote line (non-LAN) communication support″ is WAN (Wide-Area Networks). Key WAN performance considerations are contained in the following list (note that many are similar to those listed in Section 11.12.4, “LAN Line and Control Unit Parameters” on page 213):

- Data rate (throughput) improves as the frame size is increased.
- The newer IOPs (2617, 2618, 2619, 2623, 2665, 2666, 2668, and 6516) provide the IOP assist feature that is necessary for improved V3R1 APPC and TCP/IP performance.
- Fewer line turn-arounds (for example, acknowledgements) per frames sent improves performance.
- X.25 and ISDN are full duplex protocols and maximize performance significantly when data is both sent and received concurrently.

  Use the largest packet size and window size supported by the network provider and, for SNA, use MAXLANRU(*CALC). This enables the system to select an RU length that ″just fits″ into the packet size, which ensures no extra ″short packets″ are sent.

  The Performance Tools/400 reports include communication line performance information for all supported protocols in V3R6. A key packet network condition that should be reviewed is *packet congestion*. Congestion occurs when either the AS/400 system or the network is so busy that packets are not sent or received within a time limit. This is indicated when ″receive not ready″ is sent by the AS/400 system or received by the AS/400 system.
- Asynchronous and binary synchronous line protocols are significantly less efficient than the other protocols.

- For interactive environments, line utilization at 50% or less gives good performance.
- The highest possible line utilization is required for best batch data transfer performance.
- For SDLC environments, unproductive polling overhead must be minimized.

  The IOP utilization in SDLC environments does not necessarily increase consistently as the number of active workstations increases. This is because during idle data traffic periods, the IOP utilization may increase while merely polling stations. When actual data is exchanged, the IOP becomes busy processing data instead of polling.
- Multiple concurrent large data transfer jobs are necessary to achieve maximum line utilization rates for the various communication IOP maximum rated speeds.
- For LAN environments (token-ring, Ethernet, SDDI, and FDDI) examine the Resource Interval Report LAN line reports for congestion.

  If performance problems persist after verifying ″good″ configuration parameter values (frame size, LAN acknowledgement frequency, and so on), look at the communications line congestion values under the ″Congestion Local - Remote″ heading for high send or receive values.

  If the AS/400 system has high values for *local*, look for high AS/400 CPU utilization or disk busy.
- Communication line errors and resultant retries are more likely over WAN lines compared to LAN lines.

  Use the Performance Tools/400 reports to observe error retry statistics.

Table 22 on page 220 shows the best achievable WAN data transfer rates for a single program, one-way transfer for most of the supported LAN IOPs. AS/400 model 500-2141 (source) and AS/400 model 500-2142 (target) were used. IOP feature numbers are listed in parentheses.

Table 22. AS/400 WAN Lines - Maximum Throughput Capabilities

PROTOCOL (IOP)	Line Speed (kbps)	Frame Size (Bytes)	Data Rate (Kbps)	CPU Util Source/Target (%)
SDLC (2623)	56kbps	2057	53	0.4/0.9
SDLC-fT1 (2626)	640kbps	2057	541*	
Frame Relay (2666)	2048kbps	8182	1864*	
X.25 (2623)	56kbps	4096	55	0.4/0.7
ISDN/IDLC (2623)	64kbps	8192	63*	
ISDN/X.25 (2623)	64kbps	4096	62*	
BSC (2623)	56kbps	2048	52	1.2/1.9
Async (2623)	19.2kbps	2048	13	0.2/0.6

Note:

This table is a subset of a table from the *Performance Capabilities Reference* manual. The rates shown are the best rates achievable and include no database I/O in the application. The largest supported frame size and best values for SNA MAXLENRU and PACING and Frame Relay LANACKFRQ and LANMAXOUT were used.

*Depicts values which are not available for V3R6. V3R1 values are shown as a guide for V3R6.

″FT1″ is over a fractional T1 link.

Most customer applications realize lower rates than those shown as they typically perform many database I/O operations and use significantly shorter user program record lengths.

11.12.7 SNA Device/Session Parameters

From a performance viewpoint, SNA protocol exists as a layer between what an application does to affect performance and the ″maximum performance capabilities″ controlled by line speed and line type protocols such as for local area networks and SDLC. SNA support affects primarily the communication between two programs or a program and a device whereas line type protocols primarily affect communication at the control unit level. User program I/O output operations are associated more closely to SNA capabilities than to communication type capabilities.

This section focuses on performance capabilities at the ″device level″, or in SNA terminology, at the Logical Unit (LU) level. Throughout this section, LU is used to mean either a device or a program-to-program pair exchanging data. In this context, SNA does have some relationship to half-duplex versus full-duplex protocols, but the primary SNA performance parameters are Response Mode, the size of the Request/Response Unit (MAXLENRU), and the number of RUs that can be sent before an acknowledgement is required (PACING).

For APPN devices, the Class-of-Service description contains ″priority″ transmission parameters in addition to RU length and pacing values. The priority parameter, TMSPTY, can specify low/medium/high priority and has some interrelationships with the RU length and pacing parameter values. The

Class-of-Service priority is discussed at the end of this SNA device parameters section.

- On the AS/400 system, MAXLENRU determines the amount of SNA data that can be sent to a specific LU in a single ″block″. This RU (Request/Response Unit) is a block of data understood by the LUs involved and can be shorter than a line type frame or may span several frames needed to contain the complete RU. As with line type protocols, the best performance is normally achieved by using the largest RU size supported by the two communicating LUs. With two communicating application programs, there is more flexibility in supporting ″maximum values″ versus most workstation devices. The AS/400 system supports a maximum RU size of 16 384 bytes. With Release 2.2 of the 5394, a MAXLENRU of approximately 512 bytes is the default if the control unit description specifies MAXFRAME(517). For a 5494 communications controller, the MAXFRAME is 1033 bytes.

  For APPC/APPN devices, the Mode Description controls MAXLENRU values. For other device types such as 5250 displays, the AS/400 system defaults to a MAXLENRU based on the control unit MAXFRAME value supported. For other device types such as Retail (4680) devices, MAXLENRU can be explicitly specified on the device description, although a default value is supported. In most cases, the default chosen by the system is the best value.

  When the AS/400 system is functioning as a dependent LU and communicating with a System/390 host, the RU length used is specified on the S/390 system. On the S/390, RU send and receive lengths are specified in the log mode table through the MODEENT statement used for one or more LUs. Some VTAM applications such as CICS may override the VTAM MODEENT value. For example, the CICS Terminal Control Table (DFHTCT) entry for each LU can specify both send and receive RU values (RUSIZE and BUFFER).

  Dependent LU support includes the AS/400 system running Remote Job Entry (RJEF), 3270 Device Emulation, Distributed System Node Executive (DSNX), and Distributed Host Command Facility (DHCF).

  Refer to the *OS/400 Communications Management V3R1*, SC41-3406, for more device type details.

- SNA pacing specifies how many RUs can be exchanged before a response is required from the receiving LU. Once a positive response has been recognized, remaining RUs of data are sent. A high pacing value improves performance for an individual LU.

  For APPC/APPN devices, the mode description controls PACING values. For other device types, the AS/400 system takes a default based on the device type or pacing can be explicitly specified on the device description. In most cases, a pacing of seven (seven RUs sent before waiting for a positive acknowledgement) is the default.

  Most display devices run with a pacing value of zero (unpaced) because they and their attached control units have an appropriate amount of storage for the number of devices that can be attached and the amount of data that can be shown on each display.

  Batch type communications such as file transfer or remote printer output also normally choose a large pacing value for maximum throughput.

When the AS/400 system is functioning as a dependent LU and communicating with a System/390 host, the pacing used is specified on the S/390 system. On the S/390, VTAM macros such as GROUP or LU pacing is specified with the PACING parameter.

- Response mode can specify ″definite″ or ″exception″. Exception response mode means the RU can be sent and no response is needed if the data is accepted by the receiving LU. An ″exception response″ is sent from the receiving LU on an error condition. Definite response mode means either a positive or negative response is required. As you can see, anytime a definite response is required, the pacing value may be ignored; *and performance can be degraded if there is a large time delay in receiving the SNA response*.

  The AS/400 system supports both definite and exception response mode for each LU-to-LU session. The ability for the application program to control whether an exception or definite response is required on a specific output operation is variable based on the system support provided by that ″device type″.

  For example, with display devices, the system uses exception response mode for all output operations except full display file open and close, display save and restore functions, and *output only* operations. With APPC devices, exception response is used for output operations except when the DDS CONFIRM keyword is used. With retail devices, exception response is used except when the ENDGRP (End Group) DDS keyword is used.

  When the AS/400 system is running as a dependent LU and communicating with a System/390 host, the host application controls response mode for its output operations.

- For APPN devices, the TMSPTY parameter on the Class-of-Service description object can give priority to APPC data processing within each node supporting this APPN function. APPN TMSPTY assigns the ″processing priority″ of APPN-related data within the node. Once the data is sent by the node, RU length and pacing values begin to affect performance. Once the data is received by a node, that node prioritizes its processing of that data within the node according to the TMSPTY values in the Class-of-Service description being used. As a result, APPN data can be considered to have a priority while it is routed throughout an APPN network.

  Without APPN, assigning a small RU length or low pacing value to a batch transmission can favor interactive over batch as far as usage of the available communications line speed is concerned. With APPN and adaptive pacing, pacing values do not exercise the same degree of control and Class-of-Service priority should be considered. Also, when only batch transmission is being performed, low RU length and pacing values need to be increased or the batch transmission may not be able to make full use of the available communication line speed. Class-of-Service priority can be used to minimize the need to dynamically change RU length and pacing values when the operating environment can vary from stand-alone to concurrent batch and interactive.

11.12.8 Printers Attached to 5394 or 5494 Controllers

5394 microcode diskette Release 2.2 or later supports larger SNA buffers and SNA pacing values. The same applies to all microcode levels of the 5494 workstation controller. This support minimizes the number of line turnarounds, which enables improved printer throughput. Internal lab tests indicate printer throughput (pages per minute) can be improved by 15% or more.

Both the AS/400 system and the workstation controllers must be appropriately configured to take advantage of the larger values. On the AS/400 system, the appropriate line description must specify 517 as the MAXFRAME size for a 5394 and 1033 as the MAXFRAME size for a 5494. The workstation controller description can *LINKTYPE for MAXFRAME size. The printer device description should specify MAXLENRU(*CALC) and the PACING parameter should be set between three and seven (maximum).

The higher the pacing value, the more throughput for controller attached printers. A single 5394 with Release 2.2 microcode can support up to three printers using pacing values greater than three. Faster print speed may degrade interactive response time for workstations attached to the same controller. The faster the line speed, the more improved printer throughput becomes and response time degradation is minimized.

The communication line section of this document has more general considerations for frame size, RU size, and SNA pacing values.

11.12.9 Data Compression over Communication Lines

Both SNA and BSC protocols support data compression algorithms that typically can improve **batch type** transmission throughput. The sending and receiving system programs must provide compatible compression algorithm support. Communications data compression (and decompression by the receiver) may be performed by AS/400 system code at the data management or lower levels and follows standardization options when performed over SNA.

Communications data compression may also be performed at the application level, provided both the sending and receiving applications agree on the algorithm used. Regardless of whether the compression or decompression is done by the application or the AS/400 communications support, CPU utilization is a factor that needs to be considered against the entire operating environment.

The AS/400 system supports data compression for:

- File Transfer Subroutines (FTS) that include asynchronous, binary synchronous, and SNA APPC protocols:

  The FTS defaults to performing data compression at line speeds *less than 56000 bits per second (bps) and across token-ring*. FTS includes the COMPRESS parameter that can override the default conditions for data compression.

- Binary Synchronous Communications (BSC) protocol through the Add BSC Device Entry (ADDBSCDEVE) command under the Intersystem Communications File (ICF) or System/38 Create BSC File command support parameters:

  The user selects whether compression and decompression is to be used.

- Remote Job Entry Facility (RJEF) support for BSC or SNA protocol:

The host system RJE support user selects whether compression and decompression is used.

- SNA DSNX support:

  The host system Distributed System Executive support user selects whether compression and decompression is used.

- SystemView Managed System Services/400 (MSS/400):

  Managed System Services support provides change management functions based on the managing system parameters.

  MSS/400 provides some enhanced change management functions on the local AS/400 system and can be managed either by V3R6 SystemView System Manager/400 or by a host NetView Distribution Manager.

  MSS/400 may take advantage of the APPC compression support discussed in ″AS/400 APPC Data Compression Support″. However, MSS/400 is unaware of whether any data compression is being performed.

- SystemView System Manager/400 (SM/400):

  System Manager/400 gains the capabilities to perform change management support similar to the capabilities available with NetView Distribution Manager V1R5 or later. This includes distribution and management of data transfer, running remote programs, and PTFs to another AS/400 system running MSS/400 or a RISC System/6000 or OS/2 system running a NetView Distribution Manager ″client″ or ″catcher″ program.

  SM/400 may take advantage of the APPC compression support discussed in ″AS/400 APPC Data Compression Support″. However, SM/400 is unaware of whether any data compression is being performed.

- AS/400 APPC data compression support:

  The AS/400 *mode description* supports several data compression and decompression algorithm options through the DTACPR, INDTACPR (inbound data compression), and OUTDTACPR (outbound data compression) keywords. Compression techniques include:

  – The Run Length Encoding (RLE) algorithm. RLE substitutes a 1-byte or 2-byte sequence in the data stream for each repeated occurrence of the same character or character sequence. This algorithm requires no storage and less processing time than the other options.

  – Several Lempel-Ziv based algorithms:

    These compression choices are adaptive dictionary-based dynamic compression algorithms, similar to Lempel-Ziv, that compress previously seen strings to 9-bit, 10-bit, and 12-bit codes.

In general, CPU utilization caused by compress and decompress code work needs to be compared to actual results of exchanging compressed data. When systems with ″small CPU power″ are being used, the impact of overall CPU utilization needed to compress or decompress data needs to be evaluated if other application activity is present. CPU impact may result in data compression not being used.

11.12.10 Mixing Interactive and Batch Type Applications on the Same Line

When both batch and interactive applications need to be run during the same time period, the best performance can be achieved when separate lines or time periods on the same line are used for batch and interactive.

However, in many cases, the only alternative is to run both functions over the same line at the same time. Line type protocols and SNA parameters that improve batch performance have minimal impact on many interactive applications. When batch and interactive applications are run at the same time, it is common that the parameter values that are best for batch can cause performance degradation for the interactive applications. With low speed lines and with half-duplex support, this impact is at its worst. As the line speed increases, the negative interactive impact is lessened. At token-ring speeds and lines with full-duplex support, interactive degradation is minimized and batch throughput can be good.

If data is transferred primarily in one direction, the advantage of full-duplex support is minimized because most half-duplex modems currently available have short turnaround delays.

For concurrent batch and interactive on a LAN line, refer to LAN parameter values shown in Section 11.12.4, “LAN Line and Control Unit Parameters.”

For concurrent batch and interactive on an SDLC line, consider lowering the PACING values and MAXLENRU values for the batch sessions. Choosing a pacing value of 1 and a MAXLENRU of 256 for the batch sessions gives the best concurrent interactive performance. This is valid for concurrent APPC and remote workstation environments as well as for concurrent APPC display station pass-through and batch file transfer.

For APPN networks, the APPN Class-of-Service TMSPTY parameter can also be used to bias system (node) processing toward interactive transmissions over batch transmissions. For interactive applications, you can select a mode that references a high priority Class-of-Service description such as #INTER. For batch applications, you can select a mode that references a low priority Class-of-Service description such as #BATCH. If further priority is desired, use a low MAXLENRU value for the batch application mode. You can also lower the batch pacing value in the mode used for batch applications, but using seven for both batch and interactive modes is recommended for the general case. Let Version 2 APPN Class-of-Service priority do most of the work.

For APPN connections, the general recommendation is to assign a low priority Class-of-Service to batch applications and a high priority Class-of-Service to interactive applications. Use *CALC RU length and seven for pacing values on modes used for each application. This delivers reasonable prioritization of interactive applications over concurrent batch and good batch performance if batch applications run when no interactive applications are active.

Lowering the pacing value and RU length for batch applications results in favoring interactive transmission over concurrent batch, but results in inefficient batch transmission should there be no concurrent interactive applications.

11.12.11 General V3R6 APPC and TCP/IP Performance Expectations

In V3R6, both APPC and TCP/IP code paths are shortened and receive ″IOP Assist″ when the ″faster IOPs″ (2623 WAN IOP, 2617 Ethernet IOP, 2619 token-ring IOP, 6516 Integrated PC File Server, 2618 FDDI IOP, 2666 Frame Relay IOP, 2665 SDDI IOP, and 2668 Wireless LAN IOP) are used. IOP Assist includes blocking and unblocking of data and some higher level protocol TCP/IP and SNA processing within the communications IOP, removing this work from the CPU itself.

Refer to the *Performance Capabilities Reference* manual for more information on the various test results. Summary information is described in this section.

In general, APPC uses less CPU than TCP/IP on the AS/400 system. In Figure 45, TCP/IP uses 25% more CPU for small data transfers. For larger data transfers, TCP/IP uses 50% more CPU than APPC. This is because APPC is able to provide more effective blocking of data, resulting in less CPU being used.

Using large record sizes may provide a higher application data rate with a decrease in CPU time. When using larger record sizes, the CPU has less processing to perform because there are less reads and writes to transfer the same amount of data.

Note: The data presented in Figure 45 does not include CPU to access a database, only the CPU used for receiving data. When processing real transactions, the additional CPU used by TCP/IP may be only a small part of the total CPU used in a transaction, depending on the its complexity and nature. However, in a scenario where the transactions are simple in nature, the difference in CPU used when using TCP/IP must be taken into account.

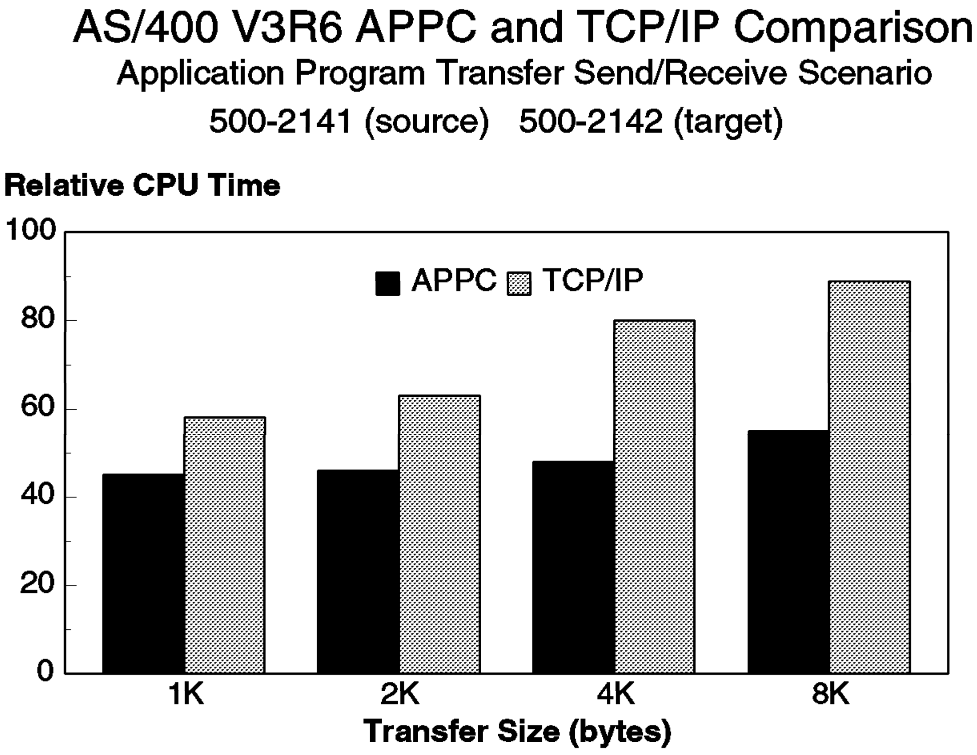


Figure 45. AS/400 APPC versus TCP/IP CPU Utilization Comparison

It is expected that in future releases of AS/400, TCP/IP will become faster, as code which is currently executing above the MI layer, is moved into SLIC.

Both APPC and TCP/IP significantly out-perform OSI in both response time and CPU usage. OSI is two to three times worse than either APPC or TCP/IP.

Many communication improvements for APPC, TCP/IP, UDP/IP, FTP, and TELNET were introduced in V3R1 and these are also available for V3R6. If you are running on a non-PowerPC AS/400 system, and running an earlier version of OS/400 (for example, V2R3), you can find more information in *Performance Capability Reference*, ZC41-8166, or *AS/400 Performance Management V3R1*, GG24-3723-02.

11.12.12 APPN Topology Routing Services (TRS Task) Overhead

APPN network node support provides powerful routing facilities that expedite routing of sessions throughout the network. However, an APPN network node may exhibit significant disk I/O operations in complex networks of many nodes. Some of the overhead can be reduced by grouping nodes into smaller subsetted networks of different network names, having few network nodes, and specifying ″Control-Point-to-Control-Point Sessions No″ on end node controller descriptions.

In some cases where a single network node is serving over 100 end nodes with a small number of switched lines on the network node, the TRS task may demonstrate high Disk I/Os at the time an end node connects or disconnects.

This overhead is in addition to and separate from session negotiation overhead at connect time.

The primary reason for the disk I/Os is that the TRS task receives a large number of Transmission Group (TG) updates that it has to process. For each TG update, TRS has to do eight to 10 disk I/O operations to its internal Topology Index.

The purpose of TG updates is to notify TRS that a TG is active or inactive (TGs correspond to controllers - you can see the TG number parameter on controller descriptions). For switched lines, the TG may be marked active prior to the link being established with the partner system by setting the MINSWTSTS parameter on the controller to *VRYONPND. A MINSWTSTS value of *VRYONPND means that a TG update indicating a TG is active should be sent to TRS when a controller is in Vary On Pending or higher state. This allows the user job to initiate the actual link activation when it is needed to efficiently utilize the line resources.

In an environment where there is an automated sequence of connection, data exchange, disconnection directed from the AS/400 network node to the end nodes such as the following example, there can be a significant number of disk I/Os in the TRS task.

In an example, assume there are 115 end nodes to be connected to a network node that has 12 switched lines. Normally the network node specifies MINSWTSTS(*VRYONPND) for each of the 115 Controllers (or TGs). Once the lines and controllers are varied on, TRS receives 115 TG updates indicating each TG is active **but not in use because the controllers are in vary on pending state when not actually connected to the network node**. When the AS/400 network node starts trying to dial out to the end nodes, the following events occur:

1. Eventually all 12 of the switched lines are in use simultaneously. TRS receives 12 TG updates (one for each of the active links) indicating each TG is active and in use.
2. However, since all 12 switched lines are in use and those are the only lines that appear in the switched line list of each of the 115 controllers, none of the other 103 controllers/TGs can be used because none of the 12 lines in their switched line list are available. As a result, since no lines are available for the other 103 controllers, TRS has to be told through 103 TG updates that each of those TGs is now inactive (even though the controllers are still in Vary On Pending state). This is necessary so TRS knows not to consider these 103 TGs when calculating a route.
3. Eventually one or more of the 12 lines becomes available, a TG update is sent for each of the TGs that were just disconnected telling TRS that the TG is active but not in use. Also, since there are now lines available for the other 103 controllers, TRS is told through another 103 TG updates that each of those TGs is now active but not in use.
4. The lines that just became available are used to connect to some of the other 115 remote end nodes, which means TG updates are sent for each TG that is active and in use and, once again, the other 103 controllers have no available lines so TRS is told through another 103 TG updates that those TGs are inactive.
5. Step 3 and Step 4 continue until the AS/400 network node gets to the point where it is no longer using all 12 lines simultaneously.

In most APPN environments, these disk I/Os do not adversely affect overall system performance.

Chapter 12. Design and Coding Tips

In addition to the tips discussed in Chapter 11, "System Performance Tuning Tips" on page 173, this section provides tips that apply more specifically to a specific system function, IBM-provided application, display workstation interfaces, and High Level Language (HLL) interfaces according to the following sequence:

- General Application Programming Tips
- Database Tips, including SQL and Query
- Compilation tips
- Display Workstation Programming Tips
- OMEGAMON Services/400 Tips
- RPG Tips
- COBOL Tips
- C Tips
- Client Access/400 and LAN Server/400 Tips and Recommendations
- LANRES/400 Tips
- ADSM/400 Tips
- ImagePlus WAF/400 Tips
- Ultimedia System Facilities/400 Tips
- DataPropagator Relational/400 Tips
- OptiConnect/400 Information

12.1.1 Overview of Program Model on PowerPC

It is important that the reader understands the program model that is used on an AS/400 Advanced System with PowerPC technology. This assists with enabling the reader to gain greater insight into the tips that are presented in this chapter.

To get the full performance benefit of the PowerPC architecture, an advanced code generation technology is required. Code optimization and code scheduling increases your application performance by eliminating redundant instructions and reducing unused cycles. This advanced technology is part of the Optimizing Translator.

To make this technology available to the existing OPM compilers, the OPM PCTs are automatically transformed into ILE PCTs. These ILE PCTs are then used by the Optimizing Translator to generate optimized RISC instructions (Figure 46 on page 230). With this, you get the benefits of advanced code generation and application investment protection automatically. The additional conversion step does affect the compile time performance of OPM programs that is discussed in the next section.

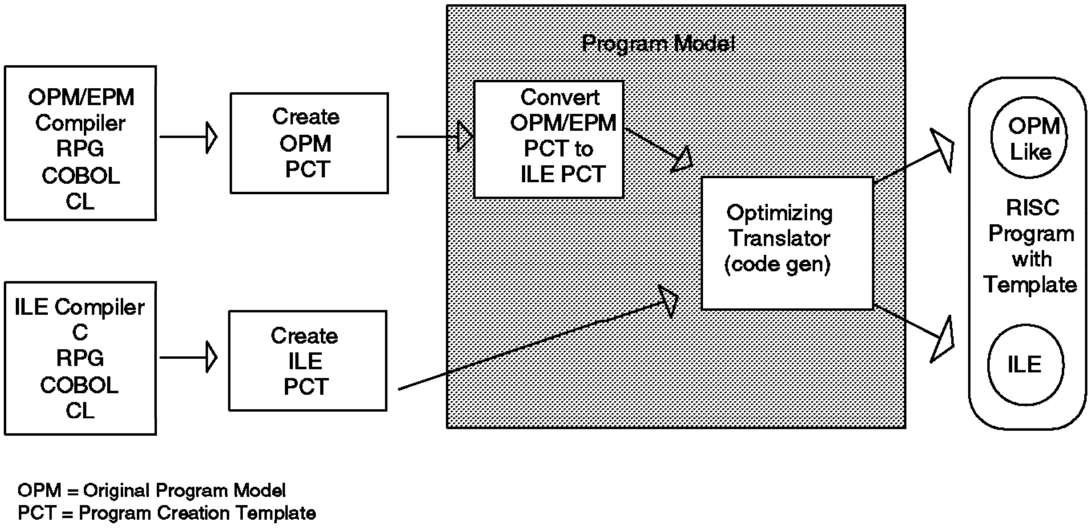

Program Model
OPM/EPM Compiler RPG COBOL CL
Create OPM PCT
Convert OPM/EPM PCT to ILE PCT
Optimizing Translator (code gen)
ILE Compiler C RPG COBOL CL
Create ILE PCT
OPM Like
RISC Program with Template
ILE
OPM = Original Program Model
PCT = Program Creation Template

Figure 46. Program Model on RISC

12.2 General Application Programming Tips

Use Odd Length Packed Fields for Numeric Data: The AS/400 system does packed decimal arithmetic. Extra SLIC instructions and extra CPU time is required to handle the extra half-byte for even-length fields. Defining numeric fields as packed decimal or moving numeric fields into a packed field within the program before calculations improves CPU utilization for frequently used fields.

Move Passed Parameters to Local Work Fields Before Use: This reduces compiler-generated code that must validate the parameter at each use.

Also, if a zoned decimal value has been passed, a local packed, odd length work field is more efficient.

If you must return the value of the parameter, copy the local work field into the external parameter before returning control.

Minimizing the number of different parameters passed improves performance. If possible, place all of the data to be passed between programs into a single parameter and use only this single ″parameter area″ between programs. This becomes more important as the frequency of calls and returns between programs increases during job run time.

Note that UNIX capabilities in V3R6 enable passing data in ″constructs″ unique to UNIX on other systems. A complete discussion of UNIX capabilities is beyond the scope of this redbook.

Variable Length Fields: Variable length field support allows a user to define any number of fields in a file as variable length, thus potentially reducing the number of bytes that need to be stored for a particular field.

Variable length field support on the AS/400 system has been implemented with a spill area, thus creating two possible situations: the non-spill case and the spill case. With this implementation, when the data overflows, all of the data is stored in the spill portion. An example is a variable length field that is defined as having a maximum length of 50 bytes and an allocated length of 20 bytes. In other words, it is expected that the majority of entries in this field are 20 bytes or less and occasionally there is a longer entry up to 50 bytes in length. When inserting an entry that has a length of 20 bytes or less, that entry is inserted into the allocated part of the field. This is an example of a non-spill case. However, if an entry is inserted (for example, 35 bytes long), all 35 bytes go into the spill area.

To create the variable length field just described, use the following SQL/400 statement:

```
CREATE TABLE library/table-name
             (field VARCHAR(50) ALLOCATE(20) NOT NULL)
```

In this particular example, the field was created with the NOT NULL option. The other two options are NULL and NOT NULL WITH DEFAULT. Refer to the NULLS section in the SQL/400 Reference Guide to determine which NULLS option are best for your use. Also, for additional information on variable length field support, refer to either the *DB2 for OS/400 SQL Reference Version 3*, SC41-4612 or the *DB2 for OS/400 SQL Programming Version 3*, SC41-4611.

Performance Expectations

- Variable length field support, when used correctly, can provide performance improvements in many environments. The savings in I/O when processing a variable length field can be significant. The biggest performance gains that are obtained from using variable length fields are for description or comment types of fields that are converted to variable length. However, because there is additional overhead associated with accessing the spill area, it is generally not a good idea to convert a field to variable length if the majority (70-100%) of the records have data in this area. To avoid this problem, design the variable length field (or fields) with the proper allocation length so that the amount of data in the spill area stays below the 60% range. This also prevents a potential waste of space with the variable length implementation.

- Another potential savings from the use of variable length fields is in DASD space. This is particularly true in implementations where there is a large difference between the ALLOCATE and the VARCHAR attributes and the amount of spill data is below 60%. Also, by minimizing the size of the file, the performance of operations such as CPYF (Copy File) are also improved.

- When using a variable length field as a join field, the impact to performance for the join depends on the number of records returned and the amount of data that spills. For a join field that contains a low percentage of spill data and that already has an index built over it that can be used in the join, a user probably finds the performance acceptable. However, if an index must be built and the field contains a large amount of overflow, a performance problem can occur when the join is processed.

- Because of the extra processing that is required for variable length fields, it is not a good idea to convert every field in a file to variable length. This is particularly true for fields that are part of an index key. Accessing records through a variable length key field is noticeably slower than through a fixed length key field. Also, index builds over variable length fields are noticeably slower than over fixed length fields.

- When accessing a file that contains variable length fields through a high-level language such as COBOL, the variable that the field is read into must be defined as variable or of a varying length. If this is not done, the data that is read into the fixed length variable is treated as fixed length. If the variable is defined as PIC X(40) and only 25 bytes of data is read in, the remaining 15 bytes are space filled. The value in that variable now contains 40 bytes. The following COBOL example shows how to declare the receiving variable as a variable length variable:

```
01 DESCR.
   49 DESCR-LEN      PIC S9(4) COMP-4.
   49 DESCRIPTION    PIC X(40).

EXEC SQL
   FETCH C1 INTO :DESCR
END-EXEC.
```

For more detail about the vary-length character string, refer to the *DB2 for OS/400 SQL Programming Version 3*, SC41-4611.

The preceding point is also true when using a high-level language to insert values into a variable length field. The variable that contains the value to be inserted must be declared as variable or varying. A PL/I example follows:

```
DCL FLD1 CHAR(40) VARYING;
FLD1 = XYZ Company;

EXEC SQL
   INSERT INTO library/file VALUES
      (″001453″, :FLD1, ...);
```

Having defined :FLD1 as VARYING, for this example, inserts a data string of 11 bytes into the field corresponding with :FLD1 in this file. If variable :FLD1 had not been defined as VARYING, a data string of 40 bytes is inserted into the corresponding field. For additional information on the VARYING attribute, refer to the *PL/I User′s Guide and Reference*, SC09-1825.

- In summary, the proper implementation and use of DB2/400 variable length field support can help provide overall improvements in both function and performance for certain types of database files. However, the amount of improvement can be greatly impacted if the new support is not used correctly so users need to take care when implementing this function.

Decimal Data Errors: The AS/400 programs detect invalid characters in decimal data fields. When this is detected, message MCH1202 is signalled to the offending program. The various high-level language compilers have different options for handling these errors, including abnormally terminating the program.

A common cause of this error is data migrated from another system that does not validate numeric field data such as the System/36. When migrating data from the System/36 to the AS/400 system, either data must be ″cleaned up″ or the processing HLL program must handle the decimal data errors. See the RPG and COBOL tips for further information.

Any HLL compiler that supports ignoring the decimal data error or converting invalid data into valid data can have degraded performance if frequent decimal data errors are signalled.

Decimal Data Performance: For ILE RPG/400 and ILE COBOL/400, use packed decimal data for best performance.

Binary Data Performance: For ILE C/400, use binary integer data for best performance.

Use ILE C/400 for Calculations: These languages can provide a performance boost for high use programs that do a significant volume of calculations.

Call Program Name Considerations (non-ILE support): *For non-ILE program calls*, use a constant or literal to name the program being called. This ensures the overhead for security and resolution of the system pointer to the called program incurred on the *first* call to that program is minimized on all subsequent calls within a run unit.

This ″saving of an already resolved pointer to a previously called program″ support also applies to variables containing program names when either RPG or COBOL performs the dynamic call. When a program reference is made with a variable, the current value of a variable is compared to the value used on the previous program reference. If the value did not change, no resolve is done to the target program. If a variable name is used in languages other than RPG and COBOL even if the same program is called again and again, the overhead to establish the pointer is incurred for each call.

Note that in V3R6, ILE C/400 also saves the resolved pointer and reuses it where possible.

Use Large Multifunction Programs (Non-ILE Support): *For non-ILE applications*, a larger program doing several functions uses less resource than calls to several programs. Good structured programming techniques are vital to the integrity and maintainability of large programs.

Simple Function Program Call (ILE support): Large multi-function programs and dynamic calls have been strengths in the application development support of the AS/400 system. However, the industry trend is toward much simpler function within a single program and calling these simpler programs repetitively within a performance measurement period, such as a transaction. This modular programming technique of small, single function programs (procedures) is also the basis for object-oriented programming application design and development.

V3R6 offers this program development support for ILE C/400, CL, ILE RPG/400, and ILE COBOL/400.

In order to take maximum performance advantage of this programming technique, you must *bind* the programs being called. This *call bound* program support or *static call* program support is discussed in the overview of ILE support in Section 3.14.6, “Activation Groups and the Integrated Language Environment” on page 55.

Note: Performance testing with V3R6 indicates RPG and COBOL ″already resolved system pointer″ support for the second and subsequent **dynamic call** processing is significantly slower than a **call bound** program within the Integrated Language Environment (ILE).

However, since the performance improvement becomes noticeable as the number of calls within a ″performance measurement period″ increases, it may not be worth the effort to convert existing applications using dynamic calls if only a single call is run within the performance measurement unit. Note also that tests have shown ″internal program calls″ (such as RPG′s EXSR, COBOL′s

PERFORM, and ILE C function inlining) to ″subroutines″ within the same program module remain faster than moving the subroutines to ″external programs″ and doing a *call bound* to that same subroutine.

The call bound support and its accompanying application development requirement leads to a general recommendation to convert a call dynamic to a call bound (static call) after it has been determined the bound program is relatively maintenance free (″debugged″) and is a critical link in achieving maximum performance.

Minimize Activation Groups Within a Job (ILE support): As discussed in Section 3.14.6, "Activation Groups and the Integrated Language Environment" on page 55, activation groups enable a logical grouping of related variables and opened files within a job. Each activation group creation is a ″mini-job startup″ on the AS/400 system and, as such, impacts performance. Therefore, the general recommendation is to keep the number of activation groups within a job to a minimum. DSPJOB options can show the activation groups in effect.

It is recommended that you consider changing the default of CRTPGM ACTGRP(*NEW) to another value such as ACTGRP(*CALLER). This minimizes the chance of causing a large number of activation groups to be created within a job. Note that ILE RPG offers an activation group value of *DFTACTGRP, which preserves the non-ILE environment while enabling the RPG program to use new RPG functions available under ILE.

The ILE C/400 Create Bound C (CRTBNDC) command supports ACTGRP(*NEW), but not *CALLER. For C, you must create a module (CRTCMOD command) and use the module in a CRTPGM command, specifying ACCGRP(*CALLER).

Service Program Considerations (ILE support): ILE offers the *service program* for packaging frequently called program/procedures into a single program for ease of maintenance.

It is recommended that the number of procedures within a service program should be kept to a reasonable number. The static storage for all of the procedures within the service program bound to an ILE program (*PGM) are initialized when the ILE *PGM is called regardless of whether the PGM directly references these procedures within the service program. This impacts performance on the first reference to the service program.

To keep static storage within a service program to a minimum, avoid large arrays within service program procedures as may be typical in some C/400 programs/procedures.

If an application needs to exchange variables with a *dynamically called* ILE program and the caller and called procedures are written in RPG IV, consider the following approach as an alternative to the normal passing of parameters. The technique uses less system resources than normal parameter passing.

- Use a service program **only** as a container for shared variables.
- For RPG IV, code the variables as **IMPORT** in the RPG procedures for the *PGMs (programs) and code the variables as **EXPORT** in the RPG procedures within the *SRVPGM (service) program.

Note: This approach also works when either the calling or called procedure is within ILE C/400.

It is the responsibility of the application developer using ILE support to make proper use of the ILE facilities within a job. ILE offers improved program modularity and performance, but misuse of ILE facilities can result in poor performance.

Controlling Activation Group Storage Within a Job (ILE Support): Just as there is concern that for storage associated with non-ILE programs that have undergone repetitive call/return sequences, the storage associated with user-specified activation groups should also be understood.

V3R6 provides the Reclaim Activation Group (RCLACTGRP) command to clean up all resources assigned to a named activation group.

Be careful when using this command. It is recommended not to use RCLACTGRP ACTGRP(*ELIGIBLE) in a production environment as this cleans up **all activation groups within the job**, irrespective of whether they are related to the application issuing the RCLACTGRP command. Consider using a call bound to the bindable API CEETREC from within an activation group when an application is ending (to delete the associated activation group) and the job is to remain active. This CEETREC API is more efficient than using the RCLACTGRP ACTGRP(activation group name) from outside the activation group.

Additional ILE information is contained under the RPG/400 and C/400 topics within this section.

Group Routines by Frequency of Use: In general, most programs are paged into main storage in 4KB blocks. Keeping active code segments together can sharply reduce paging requirements.

Make Infrequently Called Routines Subroutines: ″Dead″ code can increase program paging requirements. Moving such code into subroutines can help to keep active code together in a minimum of 4KB blocks.

Reduce Disk I/O: Disk I/O operations consume some CPU processing time and can cause waits on the disk while disk I/O for other jobs completes. So it is recommended to minimize unnecessary database operations within the program. For interactive transactions, consider 20 database operations as an excessive number.

Consider Program Observability: Using RMVOBS(*ALL) on the CHGPGM command, CHGSRVPGM command, and CHGMOD command can release a significant amount of disk space that indirectly affects performance.

For ILE programs, you have additional RMVOBS options for reducing storage (*DBGDTA (remove debug data) and *CRTDTA (remove re-create data)).

You may also use the Compress Object (CPROBJ) command to *compress observability* through the PGMOPT parameter (CPROBJ(*OBS)).

Be Careful About Removing Observability

If the application programs are to be moved to an AS/400 system that uses the CISC-based technology, they must have observability or the ″restore″ fails if the program is converted at the restore time. Also, the program does not run. It needs to have observability to be converted to CISC code. Compressed observability is sufficient to support the restore.

Reduce the Number of File Opens and Closes: Full Opens and Closes are expensive activities, not just in terms of CPU but disk activity as well. As much as possible, open the most frequently used files in a higher program invocation and use shared files (file SHARE(*YES) parameter) within the job.

The possibility of using SHARE(*YES) occurs when the application uses small modular programs where the most frequently called programs perform specific functions, and returns to a primary or ″driver″ program. In many cases, the called programs repetitively use the same database files and display record formats.

In this scenario, the primary program should open the most frequently used database files with the maximum processing possibilities (read, write, and update) with SHARE(*YES) on the files. The secondary program open file operations connect with the existing Open Data Path (ODP) that results in significant CPU savings and response time improvement.

All of the display formats used by the primary and secondary programs can be placed into a single display file that can be opened by the primary program with SHARE(*YES). This speeds up the file open by the secondary programs and can be used to permit display data to be retained across calls to several programs.

There are some things to be aware of when using shared database and display files. The secondary programs must be written with the understanding that database file record (cursor) positioning and workstation display formatting are treated by the system as if one program were using the file. For example, program PGM1 may have positioned the file SHARE to record 107 and call program PGM2. PGM2 uses shared file SHARE and causes the file to be positioned to record 23 and returns to program PGM1. PGM1 must not assume the file SHARE is still positioned at record 107. Note that if a shared file is processed sequentially for output and the file parameter specified SEQONLY(*YES), the file must be closed or the Force End of Data (FEOD) operation used for the last buffer of added records to be written to the disk.

Similarly, assume programs PGM1 and PGM2 both use record formats in file SHARED. Programs PGM1 and PGM2 must be written with the understanding of which formats may have been written to the display by the other program.

In the case of shared display files, placing more than 50 record formats in a single file can result in large Process Access Group (PAG) sizes that, in turn, result in degraded performance on storage constrained systems where many jobs are using the same display file.

Refer to *AS/400 Performance Explorer Tips and Techniques*, SG24-4781, to see how to use PEX(*STATS) option to measure CPU consumption when using SHARE(*NO) and SHARE(*YES).

If you are using the same ODP for read and update, consider using two ODPs, one for input and one for output, if you are reading the file sequentially. One ODP for input loses blocking, two allows input blocking.

Minimize Use of Distributed Data for Record-by-Record Requests: Distributed Data Management (DDM) or SQL Distributed Relational Database (DRDB) operations do not reduce CPU utilization on the local system, but rather increases CPU while reducing disk I/O on the local system. For record-by-record processing (such as filling a subfile page), keeping the ″DDM I/O operations″ per interactive transaction below six is recommended.

Using token-ring LAN or the OptiConnect/400 bus links can reduce the performance difference between local and remote access, but as the number of transactions over the link increases, the difference is more noticeable.

Increase Code Sharing: Sharing a single copy of a program by all active users is built into the AS/400 system. (All AS/400 programs are re-entrant.) Code sharing can be increased by working to ensure that as few different programs as possible are used in the system. An example of this is varying the output display or report titles for different locations within the same organization. This can be done as opposed to creating separate copies of the same program, one for each location.

Do Not Use CANCEL/FREE with SHARE(*NO): The CANCEL/FREE operation breaks the link to the ODP (Open Data Path). A second open creates a second ODP, and so on. The problem can be avoided by doing housekeeping in the called program or using shared ODPs.

Communicate between Jobs with Data Areas and Data Queues: When there is a need to communicate between jobs, the application design must make a choice between using a database file, a message queue, a data area, or a data queue.

Much less disk resource is required to update a data area or add an entry to a data queue than is needed to update a database record. The same type of savings is also realized when the receiving job reads the data.

Data queues are also more efficient than message queues, but message queues provide a wider range of function.

Consider using the user data queues (object type (*USRQ)) available through the system application programming interfaces (APIs) described in *System API Reference V3R6*, SC41-4801.

There are some disadvantages to data queues and data areas:

- Data queue or data area integrity is not as well protected as database files when a system failure occurs. Data queues do have a FORCE option to cause the data queue to be forced to disk. So you may want to use two data queues, one to exchange data and a second ″work in process″ data queue that uses the FORCE support while the message is being processed. When processing is complete, delete the entry from this in-process queue.

  If maximum protection and recovery is required, consider using journaling with a sequential database file and End-of-File-Delay (EOFDLY) support.

- As the number of data queue entries exceeds 100, performance starts to degrade.

Entries are sent to and received from data queues. When an entry is received, it is logically removed from the data queue and the space becomes available for a new data queue entry.

However, the entire space occupied by data queue entries should be considered. Although space for already received entries is reused by new entries, the ″maximum storage occupied by unreceived data queue entries″ never gets smaller. For example, assume 100 data queue entries were placed onto a data queue before any entries were received and 50 entries were received. There is now space for 50 new entries, but the total storage occupied on the system remains large enough to handle 100 entries. There is no ″compress″ function. You simply must delete a data queue and create a data queue to ″compress″ storage originally allocated.

Some applications switch between two data queues. When one data queue has reached the 100 threshold, the application switches to the second data queue while the first data queue entries are processed.

- Distributed data queue and data area support:

  In previous versions of OS/400 (V2R3, V3R0M5 and V3R1), PTFs were made available to support remote data queue and data area access through DDM support. These functions are now part of the OS/400 V3R6 base.

  – Summary of data queue support through DDM:

    The data queue APIs QCLRDTAQ (clear data queue), QSNDDTAQ (send to data queue), QRCVDTAQ (receive data from data queue), and the CRTDTAQ (Create Data Queue) command and DLTDTAQ (Delete Data Queue command support remote data queues through DDM.

    CRTDTAQ identifies the remote location and the remote data queue.

  – Summary of data area support through DDM:

    The data area OS/400 Create Data Area (CRTDTAARA) command and Change Data Area (CHGDTAARA) command and the QWCRDTAA (retrieve data area) API support remote data areas through DDM.

    CRTDTAARA identifies the remote location and the remote data area.

    The RPG data area functions are supported also.

  For reference, a list of PTFs that support Remote Data Queue PTFs is included in case a V3R6 system is communicating with a V3R1 (or earlier) system. Care should be exercised with the PTFs because they support English Upper Case only.

Remote Data Queues PTFs

V2R3	V3R0M5	V3R1M0
SF21522	SF21521	SF21555
SF19749	SF21498	SF21499
SF19748	SF21500	SF21501
SF19122	SF21254	SF21266

Remote Data Areas PTFs

V2R3	V3R0M5	V3R1M0
SF21267	SF21269	SF21270
SF19115	SF21252	SF21271
SF19136	SF21260	SF21268
SF19481	SF22045	not avail Mar 95
SF19122 (1)	SF21254	SF21266

```
(1) SF19122   This PTF temporarily disables OptiConnect400 DDM.
              After installing this PTF, apply B968529 if and only
              if you are using OptiConnect/400.
    B968529   This PTF restores OptiConnect/400 DDM.
```

Consider the Size of a Data Area: When defining an external data area, define it as large as needed but do not make it overly large for ″future use″.

Changing Date Format Considerations - Date/Time Fields: The V3R6 support for date and time fields in DB2/400 provides a number of advantages for the end user:

- Programmer productivity may be improved when an application requires calculations on date or time fields. New functions can be added more easily.
- Since the date and time data is stored in an internal format and converted on retrieval, the same underlying data can be viewed in different formats based on the needs of the application.
- Because the internal format reflects the sequential nature of time, it can be easily used to sort data in terms of sequence. For example, if a file currently contains a date in MMDDYY format, special application processing is required to sort it in YYMMDD sequence. This application processing is not needed when the date is stored in internal format.
- Some applications may achieve small savings in file size and DASD requirements since the internal formats are generally smaller than external formats.

The use of DB2/400 date/time support in many cases results in additional CPU resource being used. Generally, the increase is less than 10% but is dependent upon the number of calculations and the number of date and time fields being accessed. Time fields usually show minimal impact while date and timestamp data types may show more of an effect on performance.

Note that in terms of performance, DB2/400 date/time support is generally better than or equal to other generalized routines that support many different date/time formats. However, when compared to date/time routines that handle only specific date/time formats, DB2/400 date/time support may have higher CPU requirements.

When using date/time support in products such as AS/400 Query and Query Management, the amount of additional CPU required varies. In many instances, the impact is minimal and may even show a small reduction in CPU versus previous methods of providing this type of support. For example, report breaks on date fields under AS/400 Query in many cases provides comparable performance to using packed data for dates. However, there are certain cases where the use of date/time support can result in significant performance overhead:

- When replacing the use of zoned decimal data for dates
- When adding a result field calculations to a query (such as adding 90 days to a date)
- Report breaks on date fields under Query Management (compared to the use of packed data for dates)

Overall, DB2/400 date/time support can provide many functional advantages to user applications without a significant impact to performance. However, the user should exercise some caution when implementing this support in order to minimize this impact.

Also, you may want to consider the fact that several applications convert the MMDDYY format into YYMMDD format by multiplying the MMDDYY format by 10000.01. This uses more CPU than some other move operations. Consider using the following example (pseudo-code) for data conversion:

```
Work Areas(Length):     TMPDAT(6), TMPDAT2(4), DATE(6)

Data Field (DATE(6)) structure:    positions 1-4: MMDD1
                                             5-6: YY1
                                             1-2: YY2
                                             3-6: MMDD2

Program Move Instructions:    Move  012689  to TMPDAT
                              Move  TMPDAT  to DATE
                              Move  MMDD1   to TMPDAT2
                              Move  YY1     to YY2
                              Move  TMPDAT2 to MMDD2
```

Consider Changing a Program Table to a Database File: When performing a search on an unordered table with more than 200 entries, consider making the table a database file. Use database operations to find the ″record.″

Performance Tools: Performance Tools/400 helps you identify jobs and places within the jobs that are either consuming a large amount of CPU or performing a large amount of disk I/O operations. See the Component report and Transaction report for detailed information.

Performance Explorer: For analysis of individual program performance, Performance Explorer (PEX) contains the functions of predecessor tools such as TPST, SAM, and SMTRACE. It collects data that can help you identify the causes of performance problems that cannot be identified by collecting data using the performance monitor in Performance Tools/400.

This tool is designed for:

- Application developers
- Programmer/analysts
- Performance analysts
- Software technical service providers

who are interested in understanding or improving the performance of their programs.

More details about Performance Explorer can be found in *AS/400 Performance Explorer Tips and Techniques*, SG24-4781.

Consider Use of Query and Data File Utility Facilities: Understand what the various AS/400 query functions and Data File Utility can produce. Consider using them instead of writing a new program when the application is to select a subset of records from one or more files and complex logic is required. Some functions are part of OS/400 such as Open Query File (OPNQRYF). Other query facilities are separately priced licensed programs such as Structured Query Language

(SQL), program number 5716-ST1, and Query for OS/400, program number 5716-QU1.

Look out for any generic searches and multiple format logical files because these types of queries can adversely affect performance.

In general, writing a program that must individually process a small set of records is more efficient than using one of the query capabilities. Sorting to select the subset may be better yet. However, when there is a large number of records to process and selection logic is quite complex, using the query interfaces should be considered.

Comparison of user programming versus the various query facilities is beyond the scope of this publication. However, Section 12.5, “Queries and Structured Query Language (SQL)” on page 262 provides more query-related performance tips.

12.2.1 Application Error Handling

Improper handling of error conditions within a program can impact system performance if the error condition causes a program (job) to loop or flood message logs or the job log with messages. Consider the following situations:

- Errors happen frequently.
- If the application does not handle them, the system sometimes detects a workstation error loop and ends the job. If not, the job may loop until cancelled.
- The application should handle them or performance can suffer (along with usability and customer satisfaction).

  For multiple ILE activation groups, an unhandled exception ends the activation group. The next ILE call using that activation group incurs the overhead of starting the activation group. For additional information on *activation groups*, see Section 3.14.6, “Activation Groups and the Integrated Language Environment” on page 55.
- Design error handling into the application. Minimize application development by having one group write the error handling program and include it in the application.
- Do not ask the operator to make a choice unless it is absolutely necessary.

See “Use DEVRCYACN or Program for Workstation Errors” on page 203 for additional information on handling of workstation and communication errors.

Job performance can be affected by errors that occur within the job (such as program errors, wait timeouts, unhandled device error conditions that may cause job dumps) or by communications line/device error recovery and logging.

The application program can control many error situations if the proper error recognition logic is included in it. Many instances of poor performance can be avoided if this additional effort is incorporated into the application design and implementation.

What happens in an application that does not have error recovery procedures programmed into it? When an error occurs, it is handled by OS/400 functions and often the job is terminated. What usually happens next? The user just signs back on and continues processing at a place in the job before the error occurred. What does this scenario cost in performance? More disk I/O and CPU and memory are consumed than if an error recovery procedure handled the error and kept the job from terminating.

It is much less expensive to the end users in terms of response time and to the system in terms of CPU, disk, and memory to have the application recognize the error and perform simple recovery action rather than let the Subsystem Monitor terminate the job, put up a sign-on display, and create another job when the sign-on data is entered.

Errors are detected at all levels of the system (SLIC, OS/400, and applications). Permanent external device errors are written to the system error logs (devices, disk, and communications). Licensed Internal Code errors are logged into the SLIC log. Other errors, such a OS/400 or job errors, are logged into the QHST message queue, the system history file, the system operator′s message queue, and individual job logs.

You can determine if error logging has occurred by looking for CPU and disk I/O activity in the SLIC error logging task.

12.3 Database Tips

The following sections give many tips on how to improve the performance of your database. Some tips may help more than others. Avoid making too many changes at the one time, as it is much easier to measure improvements after making a single change.

New in V3R6: In V3R6, underneath the MI layer, the database communicates ″no record found″ and ″end of file″ through parameters rather than through MI exceptions as done in prior releases. The degree of performance improvement is dependent on the frequency of these conditions within a performance measurement period such as an interactive transaction.

Choosing Database File Processing Techniques: If the application environment allows a selection of file processing techniques (access methods), you should carefully consider which access method to use, as this can have a considerable impact on application performance.

SEQONLY and NBRRCDS for Sequential Processing: The Override Database File (OVRDBF) and Open Database File (OPNDBF) commands support blocking parameters for Sequential Only (SEQONLY). The OVRDBF command supports the NBRRCDS parameter that provides some different blocking support that improves processing in some cases. These parameters can significantly improve sequential processing performance. They enhance blocking data records and promote the operation of the adaptive look-ahead read mechanism of the system.

SEQONLY(*YES number-of-records) specifies the blocking support between the user program and OS/400 database data management support. The file must be opened for sequential input only or sequential output only for the blocking to take effect. Explicitly specifying the number of records is recommended since SEQONLY(*YES) defaults to a relatively small buffer of 4KB that minimizes advantages of blocking for large records. The maximum number of records that can be specified is 32 767.

The high-level language must support this record blocking for maximum benefit by minimizing the calls to database data management. RPG and COBOL support this blocking based on the file processing syntax in the program and physical and logical file parameters. There are several factors that determine if sequential only file processing and record blocking are actually being used for the database file. The job log indicates if sequential only file processing is actually used. The *OS/400 CL Reference*, SC41-4722, has a thorough discussion of the sequential only number of records support under the Override Database File (OVRDBF) command.

Distributed Data Management between AS/400 systems uses the number of records value in SEQONLY to determine the frequency of ″DDM function acknowledgements″. For example, assuming SEQONLY(*YES 200) is specified on the OVRDBF command that references the DDM file, a communication line trace shows that after records 2601 through 2800 (every 200 records) are sent, the target system sends a DDM acknowledgement back to the source system. This is a way of minimizing the number of line turnarounds when using DDM.

The NBRRCDS parameter on the OVRDBF command specifies the number of records moved between main storage and the disk, and may be used for either sequential or random processing. When calculating this value, divide 32 767 by your record length and discard the remainder. A large value (up to 32 767 bytes) is advantageous only when the application processes the records in the same order as they are physically stored on the disk. Using a value that is too large may degrade performance.

The largest physical block of database data that can be exchanged with the disk is 32 767 bytes. If the number of records specified in either the SEQONLY or NBRRCDS parameters is large enough, up to 32KB operations are written to or read from the disk.

The NBRRCDS parameter can be used to get the System Licensed Internal Code to bring in a specific number of records from the disk when either sequential or random processing is performed. However, it is recommended to let the system default the blocking to be the same as the SEQONLY value for number of records. If the data on the disk is physically in the order that is processed by the program, this ″bringing in records ahead of time″ can improve performance *only when the program read operation is a sequential operation (that is, not Get-by-key)*. If the user program operation scenario is a series of random operation codes, such as RPG′s CHAIN or COBOL′s READ, blocks of records read from the disk go unprocessed. If this occurs frequently, the system suffers from a large number of disk reads that are wasted because most of the records read from the disk are not processed.

Use SEQONLY(*YES nnn) if not updating or not reading a logical file that is based on one physical file and the physical data is organized in key sequence. This enables many records to be read into main storage that are available for subsequent key-based operations from the program.

Use SEQONLY(*NO) NBRRCDS(nnn) when using a logical file that is based on more than one physical file and the data is organized in key sequence. In this case, SLIC transfers data into main storage but waits for the program input operation to return data. The subsequent input operations may avoid a physical disk I/O operation when the data is already in main storage. Also, use this if updating records and data is being processed sequentially.

If you process the data in other than the physical sequence read from the disk, specific use of NBRRCDS may significantly degrade performance. Use the system default for NBRRCDS in this case. If you have sufficient main storage, consider the SETOBJACC command support for either random or sequentially processed files or expert cache for sequentially processed files.

See Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413 for more information on SETOBJACC and expert cache support.

Use of Relative Record Processing for Random Processing: This method of accessing data is supported on the AS/400 system and it is efficient when accessing the AS/400 database natively. However, in some situations such as in SQL, this may be less efficient and is not recommended.

Do Not Use Low Force Write Ratio Values: Force write ratios (specified when creating a physical or logical database file using the CRTPF or CRTLF commands respectively) interfere with the system′s tendency to do asynchronous output to the database. Journaling is, in general, a more effective way to ensure database integrity. A low FRCRATIO value significantly degrades application performance.

Note: If all of your data is placed on disks that have controllers (6501, 6502, 6512) with disk write caching support, low FRCRATIO values may result in acceptable performance.

Use Join When Processing Multiple Files: A join logical file brings records from multiple physical files together more efficiently than an application program doing multiple reads to those same files.

Note: Join files cannot be updated.

When joining files, best performance is achieved when the primary file contains the fewest number of records to be processed.

Understanding the application needs is important, as there are cases where the join range of function is not sufficient.

For additional join file tips, see Section 12.5, “Queries and Structured Query Language (SQL)” on page 262.

Use Second Normal Form (as Opposed to Third Normal Form): Normalization is a database design technique that enhances flexibility in making design changes to meet new requirements. The higher the level of normalization (5 are defined), the greater the flexibility. But the greater that the degree of normalization is, greater are the number of physical files that must be processed by the application.

Share Access Paths: Access path (index) sharing is a natural function on the AS/400 system. When a logical file is created, the system checks to see whether an existing access path meets the needs of the new file. Heavy use of logical files for select/omit processing interferes with this sharing. An access path that selects different records than another access path is not identical to the first path and cannot be shared.

Process File Adds in a Batch Job: Record additions require I/O to update the access paths. This activity can be deferred to a subsequent batch program at the cost of having a somewhat out-of-date database, or have critical access paths as update *IMMED. Use *DLY or *REBLD for other access paths. The access path maintenance parameters can be found on the CRTPF and CRTLF commands using the MAINT option.

Share Open Data Paths: Whenever possible, use shared opens of database files instead of full opens to help reduce CPU utilization. See Section 12.2, "General Application Programming Tips" on page 230 for more information.

Create Work File in Library QTEMP Once per Job: Sometimes a job needs temporary work files or work areas. Every job on the AS/400 system has an exclusive library (directory) that can be used to hold temporary objects such as files that can be used by that job. If you are coming from a System/36 and do a lot of file creations (BLDF), you get better performance if you create these files once at the start of the job and use the CLRPFM command to clear the file (or files) rather than creating and deleting the files being used. One of the advantages of QTEMP is the fact that the system cleans up and deletes all objects in QTEMP when the job terminates (or a user signs off).

The System/36 environment has a special cache-extension option to this library for System/36 environment jobs that speed up file creation (S/36 OCL BLDFILE) and deletion (S/36 OCL DELETE) if some file type requirements are satisfied. The file requirements include a physical non-source file that is not journaled and has no private authorities. The file must be program-described and be read-capable, write-capable, and update-capable. This only works in the System/36 environment.

Consider When to Use OPNQRYF Command: In batch report job streams, use OPNQRYF to subset the number of records that need to be processed. If the stream has multiple reports in the same sequence with different records needed, do one OPNQRYF, selecting all records needed for all reports. In the specific report program, select the records needed for that particular report.

Sort can be faster if no index is available and several (5000 or more) records are selected.

Consider Sorting Records for High Performance Batch Processing: In some cases, batch processing programs have used random processing I/O operations to process data sequentially. Consider doing a Sort for high performance arrival sequence processing where a copy of the data is acceptable.

Sort can be faster if no index is available and many (5000 or more) records are selected.

Minimize the Number of Logical Files: Do not create many logical files over the same physical file with small access path differences between the logical files. Remember the more access paths that exist, the longer it takes the system to update all of them when one program causes a change (record add or delete) in an access path.

See the related information on sharing access paths.

Minimize the Key Size in Logical Files: The larger the key, the higher the CPU usage is resulting from more frequent physical I/O.

Review existing logical files. If dropping the right most key field does not change the order, drop that field from the key.

Anticipate Additions to Files: For files that may exceed the default size of 10 000 records and three extents of 1000 records each, make the SIZE(*NOMAX) to avoid delays when the file is extended, or increase the values in the SIZE parameters to an appropriate value.

Reuse Deleted Record Space

Description of Function

This section discusses the support for reuse of deleted record space. This database support provides the customer with a way of placing newly-added records into previously deleted record spaces in physical files. This function should reduce the requirement for periodic physical file reorganizations to reclaim deleted record space. File reorganization can be a time consuming process depending on the size of the file and the number of indexes over it. To activate the reuse function, set the Reuse deleted records (REUSEDLT) parameter to *YES on the Create Physical File (CRTPF) command or Change Physical File CHGPF) command. The default value when creating a file is *NO (do not re-use).

Comparison to Normal Inserts

Inserts into deleted record spaces are handled differently than normal inserts and have different performance characteristics. For normal inserts into a physical file, the database support finds the end of the file and seizes it once for exclusive use for the subsequent adds. Added records are written in blocks at the end of the file. The size of the blocks written is determined by the default block size or by the size specified using an Override Database File (OVRDBF) command. The SEQONLY(*YES nnn) parameter can be used to set the block size.

In contrast, when re-use is active, the database support processes the added record the same as an update operation rather than an add operation. The database support maintains a bit map to keep track of deleted records and to provide fast access to them. Before a record can be added, the database support must use the bit-map to find the next available deleted record space, read the page containing the deleted record entry into storage, and seize the deleted record to allow replacement with the added record. Last, the added records are blocked as much as permissible and written to the file.

To summarize, additional CPU processing is required when re-use is active to find the deleted records, perform record level seizes, and maintain the bit-map of deleted records. Also, there may be some additional disk IO required to read in the deleted records prior to updating them. However, this extra overhead is generally less than the overhead associated with a sequential update operation.

Performance Expectations

The impact to performance from implementing the reuse deleted records function varies depending on the type of operation being done. The following summary shows how this function affects performance for various scenarios:

- When blocking was not specified, re-use was slightly faster or equivalent to the normal insert application. This is due to the fact that reuse by default blocks up records for disk IOs as much as possible.
- Increasing the number of indexes over a file causes degradation for all insert operations, regardless of whether reuse is used or not. However, with reuse activated, the degradation to insert operations from each additional index is generally higher than for normal inserts.

- The Reorganize Physical File Member (RGZPFM) command can run for a long period of time, depending on the number of records in the file and the number of indexes over the file. Even though activating the reuse function may cause some performance degradation, it may be justified when considering reorganization costs to reclaim deleted record space.
- The reuse function can always be de-activated if the customer encounters a critical time window where no degradation is permissible. The cost of activating/de-activating reuse is relatively low in most cases.
- Because the reuse function can lead to smaller sized files, the performance of some applications may actually improve, especially in cases where sequential non-keyed processing of a large portion of the file (or files) is taking place.

The *Performance Capabilities Reference for PowerPC Technology*, ZC41-0607, should be reviewed for specific performance test results.

SETOBJACC and Expert Cache: Consider using the Set Object Access command if you have a considerable amount of main storage available. This command loads (or attempts to load) the complete database file, database index, or the program into the assigned main storage pool.

The objective is to improve performance by accessing data within main storage and eliminate disk I/O operations.

Note: Any deleted record space is included for a database file object.

If dynamic performance tuning is active, it is recommended that you use only private pools with this command.

Consider also using expert cache support. By specifying *CALC for a shared storage pool, you ask the system to monitor object reference patterns within the storage pool and, if it detects that contiguous portions of the object are frequently referenced, to keep the identified object area in the storage pool. Normal storage management and database management algorithms are modified.

The objective is to improve performance by accessing data within main storage and eliminate disk I/O operations.

For more information on SETOBJACC and Expert Cache, see:

- *OS/400 Work Management Version 3*, SC41-4306
- Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 413

The *Performance Capabilities Reference*, ZC41-8166, should be reviewed for specific performance test results.

CCSID Support: CCSID (Coded Character Set Identification) enhancements support the dynamic conversion of data from one language to another. The support allows jobs, files, and fields within files to be tagged with an identification of the code page currently being used. For a more detailed description of this support, refer to the *National Language Support*, SC41-3103.

The main effect on performance from CCSID support is from the character data conversion required when either the CCSID of the job and the file/field do not match or when either of these CCSID values is not set to 65535. The amount of additional CPU required for this conversion varies somewhat depending on the amount of character data that needs to be converted. Since the impact of this conversion can be significant to normal database operations, users should exercise some caution when implementing this function. For example, it may be best to consider doing CCSID conversion only on fields that need the conversion done instead of all character data in the given database file.

Sort Sequence: DB2/400 sort sequence support provides application developers and end users with an easy method of producing sorted data for a particular language or culture. A set of unique and shared sort sequence tables is included on the AS/400 system. Developers can refer to sort sequences when creating applications using database, Query/400, RPG, COBOL, C, and ILE/C compilers, as well as SQL pre-compilers.

The performance of sort sequence support should be compared to the alternative methods that users have available on the AS/400 system. For example, users who want to use a different sorting sequence in QUERY/400 queries can create a translation table and specify this translation table in the ″select alternate collating sequence″ option in QUERY/400. However, comparisons of these two methods show that sort sequence support provides a noticeable improvement in performance (ranging from 5-40%) versus using the translation table method.

Users who want to learn more about sort sequences should refer to the *National Language Support*, SC41-3103.

Consider Record Locking and Waiting in the Multi-User Environment: When a database file is opened for update, the system defaults to providing a record lock when a record is read in anticipation of an update. When multiple jobs are running the same application, consider the possibility of multiple jobs waiting for a record currently locked to a job that takes a long time to process before releasing the lock. Normally, a lock is released when the locking job issues another operation to the file.

However, what happens when the locking job does not release the record in a ″reasonable amount time″ and many jobs must wait in turn for preceding jobs to release the lock? Workstation operators perceive poor response time.

Consider the following options in your application design:

- Have two files in the program that access the actual file. Open one file for input only (no record locking) and the second for update. Initially access a record through the input only file. When all processing has completed for that record, issue a read to the update file followed by an update or other operation that releases the lock on the record.

  Note: This may impact CPU utilization and may increase disk I/O′s.

- Use the ″read without lock″ support provided in RPG and COBOL. This allows only one file in the program and it can be opened for update or ″input and output″. Use similar logic previously described with the two-file approach by reading the record first with the read-without-lock operation.

  Note: Since the file was opened for both input and output, the read without lock waits on the record if another job has the record locked.

- Use a short database file WAITRCD value on the ″read for update operation″. Have the program check for this ″time-out failure″ and inform the workstation operator. The WAITRCD default is 60 seconds.
- A program that might hold the lock on a record while waiting for workstation input can be coded to protect against long operator delays. Multiple device file support can be used with the display file. If the program uses a ″write-with-invite, followed by a read-from-invited-devices″ dialogue, the display file WAITRCD can be used to timeout the workstation and run program code that unlocks the record.

Each high-level language (HLL) may or may not have unique support for explicitly unlocking a record it has locked.

If commitment control is used, the Rollback operation can be used when the WAITRCD timeout occurs on either the database or display file. Care must be taken when using this method, as rollbacks can consume considerable system resources, and should only be used as necessary.

Consider Dynamic Select When Doing Sequential Processing: In many cases, when records are reasonably distributed physically within the file, use of dynamic select and omit can improve performance when processing a subset of the records sequentially.

This is a generalization and may not deliver the best performance in some situations. Testing several different ways to process the data (such as sorting the records first or using Open Query File (OPNQRYF) command support) is recommended to determine the most efficient way to process the data in a specific customer environment.

The *DB2 for OS/400 Database Programming*, SC41-4701, discusses the possible use of dynamic select and omit techniques. However, our recommendation is to do some testing and use the approach that is best suited for a particular application environment.

Defer Access Path Maintenance with *REBLD on a Logical File: Consider using *REBLD on a logical file if the physical file member is small, the logical file is seldom used, and unique keys are not required.

Also, use defer access path maintenance with *REBLD on a seldom used logical file if the physical file member is large with many changes being made to the fields making up the key of the logical file. Unique keys must not be required.

Defer Access Path Maintenance with the Delay Option: Use defer access path maintenance with *DLY if a file is large, the number of index changes is small, and the file is seldom used. Unique keys must not be required.

Database Trigger Support Considerations: V3R6 database trigger support represents one of the most powerful features of DB2/400. Triggers are user-written programs that are called by the database manager when some change is performed in the database. The main advantage of using triggers instead of calling the program from within an application is that triggers are activated automatically, no matter which interface generated the data change. This includes local system or remote system application I/O operations on the local database (either native or SQL file I/O interfaces).

However, it is important that you take into account performance when you decide to implement triggers in your database design. Triggers are activated by means of an external call and, therefore, you need to carefully evaluate the impact on performance over the benefit of having the trigger function implemented with user program call or embedded subroutines/procedures. For example, if you are already calling your programs to do the same functions that a trigger does, we recommend continuing to use your called programs or even using ILE bound program calls rather than using triggers.

This is important because you can control the calls to the programs within your program and only do the functions when you want to. If you make the programs part of triggers, they may get called by every job that accesses the file and all trigger program processing must complete before the main application program operation is fully complete. This is acceptable if you want these functions performed on every access to the file, especially when remote systems are accessing the file on the local system.

However, poor system performance can result if certain jobs perform thousands of adds/deletes/updates. The triggers can be called for every one of these operations. This is because the triggers are defined at the file level, not the job level. In heavy batch applications or high interactive CPU utilization environments, you may not want to do this.

Other important trigger performance tips are:

- If the trigger program processing can be completed asynchronously to the main program database operation completion, consider designing the trigger program to put the function request on a data queue that is processed by a separate job.
- Minimize the number of file opens and closes within the trigger program and use shared opens wherever possible.

  If your trigger program calls other programs and they access the same files, you should try to share the Open Data Path by using the shared open option.
- Try to exit a trigger program the ″soft″ way. Avoid SETON LR in RPG, STOP RUN in COBOL, and exit( ) in C.

  This way, you should be able to leave some files open and avoid the overhead of opening them again when you get back into the trigger program. In RPG and COBOL, you should use a static variable to determine if the files need to be open or not. In C, define the file pointer as static and check for the NULL value.
- If using ILE, use the ILE default activation group wherever possible.
- If for some reason your trigger program runs in a separate activation group, remember to handle all exceptions.

  An unhandled exception terminates the activation group, closes all of the files, and causes an implicit rollback of the changes made by your trigger program.
- For SQL triggers, try to write the statements so that the optimizer chooses a reusable ODP.

Internal lab tests doing 10 000 inserts, 10 000 deletes, and 10 000 updates with two additional files accessed ″behind the user program insert/delete/write″ (in trigger programs) showed:

- For a non-commitment control environment, CPU overhead was up to 7% versus user program calls when triggers were used.
- For a commitment control environment, triggers added 17% CPU utilization versus user program calls.

  This is primarily based on the fact that the system has to place the files processed in the trigger programs into a secondary commitment cycle. (The system had to do the ″commit″ functions on behalf of the trigger programs.)

The power of trigger support can be used to **enhance performance in a distributed data environment**. Using the preceding 10 000 record insert/delete/update example, DDM files were defined for a ″program file″ and triggers were used on the remote system to do all of the update/delete/write operations on the remote system. By moving the triggers to the main file on the remote system, database operations were necessary only on the local DDM file as trigger programs and additional files and file operations were defined on the remote system. Only one DDM file was necessary and line traffic was minimized.

Consider this kind of ″trigger design″ when accessing multiple files on a remote system, especially when the OptiConnect/400 hardware and software Program Request Price Quote (PRPQ) capability is being considered.

Database Referential Integrity Support: In a database user environment, there are frequent cases where the data in one file is dependent upon the data in another file. Without support from the database management system, each application program that updates, deletes, or adds new records to the files must contain code that enforces the data dependency rules between the files. Referential Integrity (RI) is the mechanism supported by DB2/400 that offers its users the ability to enforce these rules without specifically coding them in their application (or applications). The data dependency rules are implemented as referential constraints through either CL commands or SQL statements that are available for adding, removing, and changing these constraints.

For those customers that have implemented application checking to maintain integrity of data among files, there may be a noticeable performance gain when they change the application to use the referential integrity support. The amount of improvement depends on the extent of checking in the existing application. Also, the performance gain when using RI may be greater if the application currently uses SQL statements instead of HLL native database support to enforce data dependency rules.

When implementing RI constraints, customers need to consider which data dependencies are the most commonly enforced in their applications. The customer may want to consider changing one or more of these dependencies to determine the level of performance improvement prior to a full scale implementation of all data dependencies through RI constraints.

Referential integrity is considered a requirement for industry standard compliant database support. Referential integrity enables the system to protect against mismatches in related database records according to a customer′s business rules. For example, a customer master record cannot be deleted when there are accounts payable records outstanding for that customer.

Referential integrity uses the term of *referential constraints* to mean the definition of the rules for operations valid for the dependent files, based on the constraint rule defined on the parent physical file.

A referential constraint has two major *rules*, the *delete rule* and the *update rule*:

- Delete rule:

  This rule applies when a record/row in the parent file is deleted:

  – CASCADE: The system also deletes all matching records in the dependent file.

  – SET NULL: The system sets all null-capable fields in the matching foreign keys (dependent file key that matches parent file key field) to null. Foreign key fields not null-capable are not updated.

  – SET DEFAULT: The system sets matching foreign key fields to their corresponding default value. The default foreign key value must also have a matching parent key value.

  – RESTRICT: If at least one dependent record exists, the system prevents the parent key deletion. An exception is returned.

  – NO ACTION: This is similar to the RESTRICT rule. However, enforcement and any exception is delayed until logical end of the operation.

- Update rule:

  This rule applies when a parent key is updated:

  – RESTRICT: If at least one dependent record exists, the system prevents the parent key deletion. An exception is returned.

  – NO ACTION: This is similar to the RESTRICT rule. However, enforcement and any exception is delayed until logical end of the operation.

Performance considerations when using referential integrity capabilities follow in this section. Further functional details are beyond the scope of this redbook. Refer to the *DB2 for OS/400 Database Programming*, SC41-4701, and *DB2/400 Advanced Database Functions*, GG24-4249, for additional details.

The functions actually performed by the system when processing the constraint rules are completed much faster than if they were performed with called user-written programs. This is because all functions are performed within system database code with no call/return overhead between user programs. However, just as with trigger support, the rules are defined at the file level. This is advantageous if you want the function (constraint processing) performed for every local and remote operation to the local database. It is not advantageous if you want the constraint performed only under certain applications and not others.

If you change user-written programs doing the same functions to DB2/400 referential integrity system support, remember to remove that code from your application.

Other important referential integrity tips include:

- Use foreign key fields and parent key fields with identical null attributes:

A user experiences better performance when their foreign key fields and parent key fields have identical null attributes. In fact, the non-null field attribute delivers the best performance.

Ideally, your parent and foreign key fields should not change frequently. This is due to the fact that in order to guarantee integrity, the system must verify referential integrity each time your parent key and foreign key values change. Therefore, the less your foreign and parent keys change, the system performs less referential integrity processing.

- The RESTRICT rule provides better performance since journaling and commitment control are not required.

  When a referential constraint is defined with a delete or update rule other than RESTRICT, the system has to perform some actions on the corresponding foreign keys each time a delete or an update of the parent key takes place. In order to ensure the atomicity of this operation, the system requires journaling and commitment control in some cases. If the delete rule and the update rule are other than RESTIRCT, both the parent and the dependent files must be journaled. In addition, the parent and dependent file must be journaled to the same journal receiver.

  Since the RESTRICT and NO ACTION rules cause similar rule enforcement, the RESTRICT rule provides better performance since journaling and commitment control are not required.

Null Values: DB2/400 provides support for the use of null values in any field in any file. For a more detailed description of null value support, refer to the *DB2/400 for OS/400 SQL/400 Reference*, SC41-4612 or *DB2/400 for SQL Programming*, SC41-4611.

The performance impact from using null value support varies depending on the number of fields declared as null capable and on the number of records being accessed. For example, when a user even changes only one field in a file to be null capable, there is a slight increase in the CPU resource required to either insert records into or read records from this file. The amount of the increase should be about the same whether or not the null capable field actually contains null values. Also, as the number of null capable fields in a given file record format increases, the CPU required to process each record also increases. For operations such as AS/400 Query, Query Management, and SQL/400 queries that select all of the fields from a large number of records, the impact of adding null capable fields to the file can be significant in terms of increased CPU.

Because of the potential impact, users need to be somewhat careful in which files null capable fields are used and in deciding how many fields are null capable. Although null capable fields do provide good functional advantages, performance also needs to be considered prior to using this support.

12.3.1.1 Journaling and Commitment Control

This section provides performance information and recommendations for DB2/400 journaling and commitment control.

Journaling: The primary purpose of journal management is to provide a method to recover database files. Additional uses related to performance include the use of journaling to decrease the time required to back up database files and the use of access path journaling for a potentially large reduction in the length of abnormal IPLs. For more information on the uses and management of journals, refer to the *Backup and Recovery Guide - Basic*, SC41-4304, and the *Backup and Recovery Guide - Advanced*, SC41-4305.

- The addition of journaling to an application impacts performance in terms of both CPU and I/O as the application changes to the journaled file (or files) are entered into the journal. Also, the job that is making the changes to the file must wait for the journal I/O to be written to disk, so response time, in many cases, is affected as well.

  Journaling impacts the performance of each job differently, depending largely on the amount of database writes being done. Applications doing a large number of writes to a journaled file probably show a significant degradation both in CPU and response time while an application doing only a limited number of writes to the file may show only a small impact.

- The impact to performance from adding journaling can be reduced by locating the journal receiver on a separate user ASP. Doing this generally reduces the seek time required to access the disk arms for journal I/O, which, in turn, helps reduce the impact to end user response time. It also lessens the impact to the disk arms located on the system ASP.

  When using a separate user ASP for journal receivers, it is important to consider the number of disk actuators available in the ASP. Customer environments with heavily used journal receivers located in a user ASP that consists of a single disk actuator may actually reach a limit to performance because of the high usage of this single actuator. In this case, it is better to have multiple disk actuators available in the user ASP so that DB2/400 journaling support can interleave journal entries over the multiple actuators, thus reducing contention for any one single disk arm. Doing this may result in an improvement in response time and in overall system throughput. However, it is important to note that although adding an actuator may provide a significant improvement in performance, each additional actuator added beyond this improves performance to a lesser degree. Once the utilization of the actuators is low, adding more actuators does not improve performance.

  Having two or more journal receivers located on the same user ASP and having them in use at the same time may not take full advantage of the performance gains seen by isolating a single journal receiver on the user ASP since the seek distance on the actuator increases as the journal entries are written to the two receivers.

- Tracked asynchronous I/O is used to write the journal entries to disk. Using this type of I/O allows the journal support to determine on a process-by-process basis which processes need to wait for the I/O to complete and which are allowed to continue. However, by using tracked asynchronous I/O, all I/O operations to a journal receiver now appear in performance reports as asynchronous even though the process may actually be waiting for the I/O operation to complete. This causes the Capacity Planning tools to recommend a smaller configuration than is necessary. This should be considered if a measured profile is created for purposes of future system capacity planning.

Commitment Control: Commitment control is an extension to the journal function that allows users to ensure that all changes to a transaction are either complete or, if not complete, can be easily backed out. The use of commitment control adds two more journal entries, one at the beginning of the committed transaction and one at the end, resulting in additional CPU and I/O overhead. In addition, the time that record level locks are held increases with the use of commitment control. Because of this additional overhead and possible additional record lock contention, adding commitment control, in many cases, results in a noticeable degradation in performance for an application that is currently doing journaling.

- There are instances where adding commitment control can result in improved response times for an application doing journaling. As stated before, journaling alone means that the journal entries for changes to the file are written synchronously to disk. However, under commitment control, most journal entries are written to disk asynchronously. Only the final journal entry of the commit cycle (along with any entries of the cycle that have not yet been written to disk) are written synchronously. Because of this, applications may no longer have to wait for each journaled change to be written, which can result in reduced response times. The amount of improvement depends mainly on the number of journal entries within the commit cycles (the more entries per cycle, the greater the potential for improving response time over journaling alone). For example, adding commitment control to a dedicated batch job that is currently doing journaling potentially improves the job run time if there are a large number of changes to the physical files being journaled.

  It is important to remember that the potential for improving response time by adding commitment control is also largely affected by overall system resource utilization. Environments that are showing high CPU or disk utilization or have constrained memory, in most cases, show a degradation in performance from adding commitment control because of the additional CPU and I/O required. Also, adding commitment control can result in record level lock contention between jobs, which can also affect response time. Given the number of variables involved, a test run is highly recommended prior to adding commitment control for the purpose of improving performance in a production environment.

Journaling Performance Information: Journaling and commitment control are recommended for application and data integrity protection. However, journaling and commitment control do introduce system CPU and disk I/O overhead and some understanding of this overhead is required. Neither the OS/400 Performance Monitor nor the Performance Tools/400 produce information that identify CPU resource charged to journaling. The only way to assess the overhead of journaling and commitment control is to collect performance monitor data on the same application workload without journaling and commitment control active and a second time with these functions active.

You may journal *AFTER or *BOTH database record/row entries and access paths. Commitment control requires journaling *BOTH. The more database record/row and access path changes per unit of time, the greater impact to system CPU utilization and disk I/O counts.

V3R6 commitment control received performance enhancements as V3R1 in ROLLBACK and COMMIT functions. V3R6 journaling implementation performance has been improved and makes more efficient use of the faster disks within the ASP containing the journal receiver.

If you are journaling files, you should consider placing a journal receiver into its own user ASP (Auxiliary Storage Pool). This usually gives you better performance compared to having the receivers in the system ASP because there is no disk arm stealing for I/Os not related to journaling. Another reason is that journaling I/O is done sequentially so it is done much faster when the disk arm is dedicated for the journaling functions only. If you have a moderate amount of journaling activity, you get reasonable performance at a cost of some wasted disk space.

When Journal receivers are created in a given ASP, up to 10 disk units in the ASP are used to allocate storage, which allows disk writes to be done in parallel. Previous to V3R1/V3R6 when more than 10 units were available in an ASP, journal receivers simply used the first 10 configured units.

V3R6 journal receivers are spread over the 10 fastest disks within an ASP. Performance can be improved significantly when these fast disk units or their IOPs contain write cache. Normally, processes for writing journal entries must wait for an I/O write to complete before guaranteeing that the journal information is safely on disk before files are actually modified. But when using write cache, the information simply must be copied to an internal buffer, after which the process can continue. (A battery backup preserves the information if the system crashes.) This can improve performance by 20% to 30%. An example of such devices include the 9337-2xx/4xx disks and the 6502/6512 internal RAID controller attached disks.

The V3R6 Performance Tools/400 Component Report includes a ″Database Journaling Summary″ set of statistics. These statistics include counting the number of journal entries (″deposits″) written, the number of ″bundles″ (blocks of deposits) written to journals, and, for System Managed Access Path Protection (SMAPP, discussed in "Consider System Managed Access Path Protection (SMAPP)"), the minutes of access path recovery time exposure if access paths are not being journaled.

Bundles typically contain many actual journal entries/deposits, thus minimizing actual disk I/Os for journaling. By observing the numbers of journal bundles and entries written for either user-specified journaling or System Managed Access Path Protection journaling, you can assess the disk I/O impact of journaling over time.

You still cannot directly identify the impact on CPU by journaling but you can record the number of bundle writes and deposits and compare them over time to see if the journal disk I/Os are increasing per performance monitor measurement time period.

For more information on database journaling and commitment control, refer to *Backup and Recovery - Advanced V3R6*, SC41-4305.

Consider System Managed Access Path Protection (SMAPP): History from AS/400 systems recovering from abnormal system failures has shown typically 75% of all recovery time overhead can be charged to recovering access paths (indexes) that were in use at the time of failure. This recovery is needed to ensure the integrity of user and system access paths in use at the time of failure.

AS/400 database specifies access path rebuild command options to journal access paths (STRJRNAP command) and control the sequence of access paths to be recovered (Edit Rebuild of Access Path (EDTRBDAP) command). This requires the user to specifically identify the access paths to be recovered and maintain this list as files and associated access paths are added to or removed from the system, such as when a new application is added or an existing application is modified.

The database journaling support provides sufficient support for those customers who take the time to fully implement a recovery plan. However, customer history has shown most customers, especially those with smaller AS/400 models, do not take the time to manage access path recovery.

With SMAPP, a *target recovery time*, applicable system wide or separately for each auxiliary storage pool (ASP), is provided. The user has options for not performing automated journaling of access paths (*OFF), estimating recovery time but not automatically journaling access paths (*NONE), or specifying the maximum amount of time (*MIN or time-value) that access path recovery should take during IPL. SMAPP tries to ensure that all required access paths are rebuilt during the specified time frame.

The user can combine explicit journaling with system access path journaling under SMAPP, where SMAPP targets access paths that are not already started by the user.

Note: SMAPP should not be viewed as a substitute for explicit journal management.

For access paths being journaled under SMAPP support, the access path journal entries are placed into internal journal receivers unless the associated file is also being journaled. In that case, the SMAPP journal entries are placed into the associated journal receiver. These SMAPP entries cannot be accessed by OS/400 users. On user-defined journal receivers, a SMAPP entry appears as a missing entry sequence number.

Remember, any journaling activity adds to system CPU utilization and disk I/O versus no journaling. Journaling also uses auxiliary disk storage. However, SMAPP is an efficient implementation of journaling and can reduce overheads by grouping information in the SMAPP journal in “bundles” of approximately 32KB at a time before the data gets written asynchronously to the disk drives. Note that there may occasionally be a slight perceptible delay between the customer′s confirmation of the change made to the database and the corresponding image confirmed in SMAPPs log.

Refer to the *Backup and Recovery - Advanced V3R6*, SC41-4305, for more details on SMAPP support.

SMAPP performance considerations include:

- SMAPP causes increased disk activity:

  SMAPP causes increased disk activity compared to no journaling of access paths. This increases the load on disk I/O processors (IOPs) disk I/O processors. Because the disk write operations for SMAPP are asynchronous, they do not directly affect response for a specific transaction. However, overall response time may be affected because of increased disk activity.

  It is not advisable to specify target access path recovery times for both the entire system and individual ASPs. If you specify both, the system does extra work trying to balance the overall system target with individual ASP targets.

- Expect SMAPP CPU utilization to be an additional 1%-6%:

  SMAPP CPU utilization is lower than user-specified access path journaling, but can reach up to an additional 6% where there is an excessively high frequency of access path changes (record added, deleted, and key fields updated) and a short recovery time specified.

  The shorter the target recovery time, the greater the performance overheads. A maximum value of 1440 minutes (one day) is supported. Any system overhead incurred by SMAPP needs to be balanced against the reduction in IPL time spent recovering access paths as a result of an abnormal termination.

- SMAPP is set to ″on″ with a recovery time of 150 minutes after V3R6 is installed:

  Lab tests indicate 150 minute recovery time is a reasonable balance between run-time CPU utilization overhead and IPL recovery time.

- Set SMAPP time value to *NONE if previous release CPU utilization is already approaching 95+% and interactive response time or job throughput rates or run-time limits are approaching unacceptable values:

  You may also specify *OFF (no automatic journaling of access paths, no recovery time exposure estimating) or *NONE (no automatic journaling of access paths, but estimate recovery time exposure) if you observe high CPU utilization and understand your environment is making lots of changes to access paths.

  Regardless of the recovery time target values, you can use the Edit Recovery of Access Paths (EDTRCYAP) command or Display Recovery of Access Paths (DSPRCYAP) command, or the V3R6 ″Database Journaling″ heading of the Performance Tools/400 Component Report to review SMAPP-related information. By observing the exposure times, the user can determine if SMAPP recovery time should be set to a valid time value or the user should explicitly journal an access path.

- SMAPP recovery time has a special recovery time value of *MIN:

  *MIN protects all of the access paths for the entire system. With this value, the user does not have to know what the actual minimum recovery time is for the system, but indicates that the system should always provide for the fastest access path recovery.

  However, *MIN potentially causes the highest run-time SMAPP CPU utilization because of its ″shortest″ recovery time meaning.

Detailed statistics for both user specified journaling and SMAPP provided journaling can be found on the ″Database Journal Summary″ section of the Component Report. A sample is shown in Chapter 8 of *Performance Tools/400 Version 3* manual, SC41-4340.

V3R6 One-Time Cross Reference File Conversion: As discussed in Section 11.1.6, “Database System Cross Reference Job (QDBSRVXR)” on page 175, the installation of V3R6 builds a system wide catalogue of all database files. When coming from a previous release other than V3R1, a one-time build process occurs to load the system-wide catalogue. In addition, restoring a previous release SQL collection requires this same build process. System job QDBSRVXR actually performs the build with assistance from a QDBSRVnn system jobs.

Note: As shipped with V3R6, job QDBSRVXR runs at priority 0 and the QDBSRVnn jobs run at varying priorities from 9 to 52. If these jobs are busy loading cross-reference information during normal production hours, production jobs may experience performance degradation.

The post-installation portion of the conversion may cause certain system functions that use the system database cross-reference to be delayed for an extended period, and sometimes to fail. A STRSQL request may remain in a wait-status for several minutes before finally starting with an error and a message about the database connection defaulting to *N. Requests to modify the system relational database directory will fail. These problems are caused by the QDBSRVXR job being too busy to service requests in a timely manner. The problems can be resolved by waiting for the QDBSRVXR job to return to a WRKACTJOB command DEQW (dequeue wait) status after it has remained in a near-constant RUN (running) status. This is typically only a concern immediately after a V3R6 installation, a catalog restore, a RCLSTG, or a database reclaim request (program QDBRCLXR) when the reference file build has not yet processed all files/tables.

You can force the conversion to complete by placing the system into restricted state after V3R6 IPL and calling V3R6 program QDBRCLXR (database cross reference reclaim) as follows:

```
QSYS/CALL QSYS/QDBRCLXR
```

This cross reference information build was made to enable faster access from SQL catalog files and to meet SQL standards.

The information being maintained is stored in several database files QADB* in library QSYS. The four files (QADBFDEP, QADBPKG, QADBXRDBD, and QADBXREF) have existed before V3R1 and V3R6. Starting in V3R1 (and being retained in V3R6), the four additional files (QADBIFLD, QADBKFLD, QADBCCST, and QADBFCST) also exist. These files are used as physical files for SQL VIEW files that are created for the SQL CATALOG.

The full description of this cross reference information conversion is contained in an ″information only″ APAR II08311.

12.4 General Design Considerations

This section gives you an idea of the relative system efficiencies associated with certain functions. All OS/400 commands, modules, and MI-SVLs use system resource. It is the responsibility of the programmer to know which are most system efficient.

Note: The system resource used by performing a certain function within a specified time frame and its ramifications must always be fully understood.

For example, an order entry application may be the most critical application for an enterprise. It is important that it run efficiently with as minimal response time as possible. Because of its critical nature to a business and its frequency of use, a minor improvement in efficiency results in a major reduction of system resource use for all other applications. Alternatively, a batch program run once a month at night may not require the same attention to the use of system resources. The amount of resources used probably does not affect other users.

Consider the following partial list of general guidelines when designing AS/400 application programs and databases. For complete design guidelines, please contact your IBM marketing representative, or authorized IBM business partner.

Database Design Considerations

- Having a large number of database indices defined in your database design can affect performance.
- Applications that open a file in several programs in a job′s stack should use files created with SHARE(*YES). The file should be opened by a driver program. All subsequent open/close operations use much less CPU time.
- File level conflicts occur most frequently when a physical file has a large number of indices (or access paths) defined for it. For example, whenever a record is added to a physical file, an exclusive seize is held on the file until all indices have been updated. While the logical files are being updated, processes requesting access to the physical file must wait.
- Record-level locks can also be a problem but to a lesser degree. An example is when a data area is used with an application, and a number of workstations are trying to update the control record at the same time.
- The use of the FRCRATIO parameter on a database file is unnecessary in most cases. Whenever there is a critical need for database changes to be written to disk immediately, journaling should be used.
- When defining numeric fields in a database record, the application designer should always try to specify packed decimal fields with odd lengths. Internally, when doing arithmetic on even length packed decimal numbers, that number is converted to an odd length, operated upon, and converted back to an even length. The conversion process in this case consumes much more CPU than the actual arithmetic operation.

Application Design Considerations

- Whenever possible, use shared rather than full opens. It can be shown by benchmarking that full opens use approximately 10 to 15 times the CPU resources of shared opens. Also, files should be opened for sharing at the highest level possible. Therefore, in the highest level program, open only those files that are used most frequently by lower-level programs.
- Call statements should specify the program name as a literal rather than a variable. When the constant form is used, the overhead required is performed only for the first call of a program. When a variable program name is used, the address of the program must be resolved each time a call is made.

Program Design Considerations

- One factor of the time spent in an activity level is the number of database I/O operations per transaction. The time taken per database operation can vary depending on disk subsystem configuration. As a guide, the time taken per database I/O can be estimated by dividing the number of synchronous I/O′s per transaction by the number of logical database I/O′s per transaction and multiplying the result by the average access time for the DASD subsystem. The raw data can be obtained from the Performance Tools System Report.
- Application designers should be aware that each separate field or literal constant definition on a display causes overhead when written to or read from a workstation. Combine constant fields wherever possible.
- When a search of an unordered table with more than 200 entries is required, use a file instead. (Unordered tables must be searched entirely each time they are searched.)

- Numeric editing is a relatively costly operation in terms of performance and should always be minimized:

  Each read or write causes considerable system overhead. An example of minimizing I/O operations is to design formats to fill the entire display.

- Minimize error handling in programs by using the validity checking provided by DDS whenever possible. Use the ERRMSG of DDS to eliminate formatting and putting an error message to the workstation.
- In general, the best way to build a program on the AS/400 system is to combine functions from several programs into a single program. A significant number of external calls can affect performance. Also, the overhead of calling a CL program from a high-level language program is higher than the normal call/return. This type of call should be minimized. Refer to Section C.3, "AS/400 IBMLIB" on page 394 for a description of the PRTTRCSUM command. This command is useful for summarizing the output of the Trace Job (TRCJOB) command, which helps in identifying programs that can be combined to reduce external calls.
- Avoid creating a CL program with the attribute to log all CL commands. You should only log CL commands when you are testing a job, not when running in a production environment.
- Specifying DFRWRT(*YES) when using the CRTDSPF command may result in improved performance, particularly if your count of WSPUT modules is greater than WSGET modules in the Module Report.
- Where possible, avoid using the override attribute (OVRATR) and override data (OVRDTA) in the DDS for all display files.

12.5 Queries and Structured Query Language (SQL)

In general, V3R6 provides equivalent database performance to that of V3R1 on systems of equal relative performance ratings. In addition, there have been several DB2/400 enhancements made in V3R6 that help users obtain better performance.

12.5.1 Enhanced Index Support for DB2/400

The index support for DB2/400 on V3R6 has been significantly enhanced to support larger indexes and to reduce index seize contention. Now, rather than a size limitation of 4GB, each index can be as large as 1TB. This limit is related to the size of the index and not the number of entries.

To take advantage of the increased index size capabilities, a new parameter (ACCPTHSIZ (Access Path Size)) has been added to the Create Physical File (CRTPF) command, Create Logical File (CRTLF) command, and Change Physical File (CHGPF) command. Its default value is *MAX4GB. If you want to allow growth beyond 4GB on a keyed physical file that already exists, you can use the Change Physical File (CHGPF) command specifying ″*MAX1TB″ for the ACCESS PATH SIZE parameter. If you want to change a logical file to the larger limit, delete the existing logical file and create a new logical file specifying *MAX1TB for the ACCESS PATH SIZE parameter. When creating new files, physical or logical, the default is *MAX4GB. You must specify *MAX1TB if it is required. It is recommended, though not required, that all access paths for a file be of the same type.

Note: The procedure previously described does not work for files that have UNIQUE specified. For these files, you must create a second file with the Access Path Size specified at *MAX1TB and copy the first file to the second. The first file can be deleted and the second file renamed.

In addition to the capability to create larger files, the algorithm used for seizing indexes has been enhanced. Indexes are now seized at the index page level. Therefore, for those workloads where there is a high-level of concurrency on a particular file or access path, the new algorithm significantly reduces the contention resulting in significant performance improvements. This is particularly applicable to multi-processor systems and indexes with a high number of records being added. To take advantage of this change, the index must be created/changed as previously specified.

Most customers are best served by specifying the default of *MAX4GB for the ACCESS PATH SIZE parameter. This, in general, provides better performance. Also, if files may be moved to a prior release, the index may need to be rebuilt or saving the file does not work if the index was created with *MAX1TB.

It is recommended that *MAX1TB be specified only where it is needed to allow for the larger file size or where there is high contention on access paths. If an index is changed to *MAX1TB and there is no high contention on the index, it may result in additional overhead. One measurement on a uni-processor system where this was the case resulted in a slight (less than 5%) slowdown. The amount of overhead depends on, but is not limited to, the number of files, size of files, and access patterns.

12.5.1.1 How to Determine When to Switch to *MAX1TB for Performance

To determine if changing to a *MAX1TB access path improves performance, you need to collect data using the Performance Monitor. This shows if the contention on your system is at a level where changing the maximum access path size may help performance. Performance data can be collected using the Start Performance Monitor (STRPFRMOM) command. When issuing this command, the interval value should be changed from 15 minutes to 5 minutes and the trace data collected should be changed from *NONE to *ALL. Data should be collected for at least 30 minutes and should be collected during ″peak″ activity. It is important to note that when the Performance Monitor is ended, the trace data is automatically dumped to DASD and performance is degraded while this data is being transferred. This can be avoided by specifying Dump Trace *NO option when starting the monitor and at a later time, when the system is not as busy, issuing the DMPTRC command.

If the number of Seize Conflicts on the Component Report (created by issuing the PRTCPTRPT command or using the menu options from the PERFORM menu) is greater than **140 per second**, you can, in many cases, benefit from changing to files with the ACCESS PATH SIZE parameter set to *MAX1TB. The method to determine which files should be changed is discussed in the next paragraph.

Once the trace data has been saved (either when the monitor ends if Dump Trace *YES was specified or after issuing the DMPTRC command), a transaction report should be printed by either issuing the PRTTNSRPT command or using the menu options from the PERFORM menu. The SUMMARY OF SEIZE/LOCK CONFLICTS BY OBJECTS section of this report shows the number of seize conflicts and the number of lock conflicts by object. This list shows conflicts on both data spaces (that is, file data) and data space indexes (that is, access paths). If the majority of seize conflicts are on one or two data space indexes, those are the candidates for the larger access path size parameters. The files most likely to be in this range are those that are accessed by different users at the same time, some for inserting into the file, some for updating records in the file, and some for reading records in the file.

For more details on how to perform the preceding functions and what the reports contain as well as how to interpret the data, please consult the *Performance Tools/400 Version 3*, SC41-4340.

12.5.2 DB2/400 SQL and Query Information Improvements

In V3R6, there have been several enhancements made to improve performance for SQL queries. In addition, more information is now available to help users analyze performance for all queries.

1. V3R6 now has more information provided to query users to help them analyze and improve the performance of their queries:

   - Index advisor messages have now been added to the messages that are generated in the job log when running a query in a job in DEBUG mode. These messages indicate how an index can be constructed that is optimal for the performance of that query. The information provided is generally most useful for queries that involve a single file or for the primary file in a join query.
   - There is a new database performance monitor function available through the Start Database Monitor (STRDBMON) command. This monitor provides detailed information on all DB2/400 queries, such as CPU, I/O, elapsed time, description of the query, and so on. The information is placed in a database file where it can be readily queried. The data provided by this function can provide valuable information for performance analysis of any DB2/400 query.

     > **IMPORTANT**
     >
     > Care should be taken when running this command, particularly on large systems performing many SQL commands. If it is run for extended periods of time, large amounts of disk space may be consumed. For each query, up 10 records of data are written to disk and each record is 528 bytes long. Also once running, there is no easy way of finding if it is running as it does not show up in WRKACTJOB.

     For more information on the database monitor, refer to the *DB2 for OS/400 Database Programming V3R6*, SC41-4701.

2. Support has been added to allow more types of join operations such as outer joins and exception joins. Although users could previously construct queries involving UNIONs for outer joins, this was often cumbersome to do. The new SQL join syntax now gives users the ability to easily code these types of queries, often with significantly improved performance versus the previous alternative methods. For more information on the new syntax, refer to *DB2 for OS/400 SQL Programming V3R6*, SC41-4611.

3. The new join syntax for SQL now also gives users the ability to specify the order in which they want files to be joined. This can help improve performance for join queries where the optimizer is not choosing the optimal

order in which to do the join. More information on this is available in the previously mentioned *DB2 for OS/400 SQL Programming Version 3*, SC41-4611.

4. Prior to V3R6, all SQL cursors within a given program/module operated under the commitment control level specified for that program when it was compiled. Now, however, a new WITH clause has been added to the SQL SELECT statement to allow specific cursors to run under the desired level of commitment control. For example, if an SQL program is running with a commit level of *ALL but there are read-only cursors in the program that do not need any commitment control, you can add the WITH NC clause to the SELECT statements to have these cursors run under a level of *NONE. Other levels that can be specified are UR (for *CHG), CS (for *CS), and RS (for *ALL). In many cases, a significant improvement in performance can be realized when this type of change is made to cursors that previously had been running under a more stringent level of commitment control than was necessary.

5. Support has been added to the ALTER TABLE SQL command that gives users the ability to easily add new fields and delete/change existing fields in any database file. Although users could do this in the past by deleting the old file and recreating it with a new format, this also meant that any views and indexes over the file had to be rebuilt as well, which could take a long time to complete. With the new support, the existing database file is copied to another file with the new format, and the indexes over the file are not rebuilt as long as no key fields are being altered. Although it still may take a while to do this copy, altering a database file′s format with ALTER TABLE should be considerably faster than what the user had to do previously.

6. Prior to V3R6, a UNION ALL operation that did not specify an ORDER BY always generated a temporary file containing the results from each of the SELECT statements. This also meant that the ODP for this UNION was not reusable, which resulted in a full open and close each time the UNION was run. In V3R6, this type of operation now operates with ″live″ data, that is, the results from the second SELECT are not read until all of the rows from the first SELECT have been read. This change, in most cases, results in significantly improved response times for the first rows returned from the UNION since these rows can be returned immediately without having to wait for the entire temporary file to be filled with results from all of the SELECTs. Also, SQL is now able to make the ODP for this operation reusable, which also improves response time significantly.

7. In previous releases, the ODP for a cursor that contained a LIKE clause with a host variable mask was not reusable, which meant a full open was required each time the query was run. In V3R6, the ODP is reusable if the value in the host variable mask is of the form ′XXXX%′ and the NUMBER of constant characters in the mask stays exactly the same between each run of the query. In the case shown here, the contents of the XXXX constant part may be changed, but the number of constant characters (4) must remain the same and there cannot be anything else in the mask. If these rules are adhered to, users can significantly improve the performance of this type of SQL query.

8. Queries such as UNIONs, subqueries, and joins may in some cases specify the same view in each SELECT specified in the query. Prior to V3R6, running these types of queries resulted in the view being evaluated multiple times, once for each time it was specified. In V3R6, the view in cases such as this is evaluated once and the results are used by each SELECT in the query.

This change can result in a noticeable improvement in performance for this type of query, particularly if each evaluation of the view is costly.

12.6 DB2/400 SQL

DB2/400 Structured Query Language (SQL) support provides the user with an additional means of accessing data within an OS/400 relational database. This support provides several advantages in terms of flexibility, productivity, and portability between various database platforms. However, prior to using SQL, users should also consider what level of performance they can expect when using this product. This section provides general information on the performance of SQL to help users better determine what this level of performance is.

This section is not intended to be a complete guide to SQL performance. For many users, other items of interest include the SQL optimizer, performance tips and techniques, database design, and so on. It is important that users take advantage of SQL performance tips and techniques as much as possible when writing an application using SQL. In particular, it is important to properly construct and use indexes to provide the best overall performance for SQL. This, along with many other tips and techniques, can be found in the sources listed in Section 12.6.1.3, "Additional Sources of Information" on page 267.

In V3R6, enhancements have been made to SQL that help many users obtain an overall performance improvement. For a description of these enhancements, refer to Section 12.5.2, "DB2/400 SQL and Query Information Improvements" on page 264.

12.6.1.1 Performance of SQL versus Native DB

When current users of AS/400 native language I/O (that is, COBOL/400, RPG/400, and so on) are considering using SQL in their applications, one key item that needs to be considered is the difference in performance that is to be expected when making this change.

This section does not provide detailed information on the difference in performance between native and SQL for specific types of I/O operations.

It is difficult to predict how SQL compares to native DB access for a given application. Generally, SQL uses anywhere from 10% to 30% more CPU than native, although this may vary considerably depending on the type of operations being done. For example, SQL uses considerably more overhead than native when operating on one record at a time, such as an OPEN-FETCH-UPDATE WHERE CURRENT OF-CLOSE sequence. However, for more complex operations or for operations involving a lower number of SQL statements, SQL, in many cases, shows equal or better performance levels than native.

The difference between SQL and native performance is usually in terms of CPU (the amount of I/O that occurs is generally about the same for native and SQL for similar functions). On systems where the CPU utilization has not reached the knee of the performance curve, a difference in CPU of 10% to 30% per transaction does not result in a large difference in response time. Beyond the knee, however, the response time difference may grow considerably.

12.6.1.2 Other Performance Notes

- Generally, the use of SQL results in significantly increased memory requirements when compared to similar native DB operations. This is mainly due to the additional internal structures and program automatic storage required by SQL to maintain optimum performance levels, as well as the fact that SQL cannot share ODPs across or within applications as native DB can. Remember that the extra memory required can vary widely from application-to-application and is mostly dependent on the complexity of the application. Simple applications involving only a small number of I/O operations may require little additional memory for SQL, but for complex applications involving many I/O operations, the difference in memory requirements between native and SQL can be significant. When using SQL, users should monitor memory utilization using the AS/400 Performance Tools to better determine if additional memory is required.
- Some SQL users may notice an increase in the amount of auxiliary storage used once their applications begin to run. This is again largely due to the number and complexity of the ODPs and other internal structures that are maintained by SQL for each individual user for optimum performance. This additional storage requirement is only temporary, that is, when the user′s job ends, the storage returns to normal levels. For systems where auxiliary storage usage is already high, some evaluation of the number of active SQL users and their storage requirements may be needed prior to any large scale implementation of a given SQL application.
- If memory and auxiliary storage is a concern, there are ways to reduce consumption of these resources for SQL applications. The following methods can be used:
  - In a given SQL program, combine any duplicate or same SQL statements into one statement in an internal procedure, and call that procedure as needed. This helps reduce the number of ODPs for that program.
  - If an internal procedure containing SQL statements is duplicated in several different SQL programs, the number of ODPs can be reduced by placing this internal procedure into a separate SQL program and calling that program as needed. However, the user needs to be careful when doing this to ensure that the ODPs in this common program are reusable across different invocations of the program. Also, since external calls cause some performance degradation, this also needs to be considered prior to implementing this type of change.
  - Some user applications ″pre-open″ all of their SQL ODPs in order to avoid full opens when the SQL statements are issued. Although this provides good performance in many cases, there may be ODPs that are pre-opened but rarely used. If this is true, the developer may want to be more selective about which ODPs are pre-opened and which are left to be opened when the SQL statement is first issued.

12.6.1.3 Additional Sources of Information

There are several other sources of information available that the user should obtain to gain a better understanding of SQL and OS/400 database operations, as well as understanding how to properly code SQL functions in order to optimize performance. The following list contains these sources:.

- *AS/400 Performance Capabilities Reference*
  - V3R2 edition is ZC41-8166.

  - V3R7 edition is ZC41-0607.

- *Quicksizer* support for SQL:

  Available on HONE to help users size their system for using SQL.

- *System Handbook*, GA19-5486-14
- *DB2 for OS/400 SQL Programming V3R6*, SC41-4611
- *DB2 for OS/400 SQL Reference V3R6*, SC41-4612
- *DB2 for OS/400 Database Programming V3R6*, SC41-4701
- *OS/400 Data Management V3R6*, SC41-4710
- *DB2 for OS/400 SQL Call Level Interface V3R6*, SC41-4806

V3R6 query support (SQL, Query/400, Open Query File support, and so on) contains several performance enhancements. Some enhancements are to be used while ″debugging″ performance of a query application. Most are internal implementations that applications can take advantage of with no changes. In some cases, re-creating the SQL program/package or some source statement SQL change is required to make use of the improved support.

A thorough discussion of query performance tips is beyond the scope of this redbook. Some tips are presented herein but they should not be considered a complete set of documentation to achieve maximum query performance. Achieving maximum AS/400 query performance requires a detailed understanding of IBM documentation and the ability to analyze the query source statements and the database record/key distribution unique for each customer′s environment.

Generally, if the query processes a single file of 10 000 records or less, performance is acceptable to customers. Processing files with more than 10 000 records and join queries and multiple jobs using these queries usually require ″design for performance″ considerations and sometimes performance problem analysis.

The Performance Tools/400 can provide CPU utilization and disk I/O counts for jobs that use queries. To determine the logic decisions made by the AS/400 query components, such as the *query optimizer*, you must review the output of the following tools:

- Print SQL Information (PRTSQLINF) command:

  This command prints the access plan determined at SQL program or package creation. You can examine what indexes are used or if it has already been decided that temporary indexes are to be used. If you understand the customer database and key field population, you can already determine if the access plan is likely to result in good performance.

  Note that when the query actually runs, the query component may detect a change from compile time analysis of the file/table and decide to create a new access plan at run time.

- Place the job into ″debug mode″ and review the job log messages produced by the query components, including the optimizer.

  These messages determine which access method techniques are being used at run time. Analysis of these messages may necessitate a review of the query statements and their selection syntax or the creation of an index that matches the select criteria.

Note that indexes created after V2R3 contain statistics on unique key values for up to the first four keys. These statistics can assist the optimizer′s decision making process.

- Change Query Attributes (CHGQRYA) command:

  The CHGQRYA command applies to all queries run by the job. The QRYTIMLMT parameter can be used to specify a time limit in seconds. A message is issued by the optimizer if the optimizer estimates that running the query takes longer than the specified time limit. Message CPA4259 is issued that indicates which access plan is used. CPA4259 requires a cancel or ignore response that can be responded to by an operator or automatically through a default reply or an entry in the system reply list.

  The following examples show a system reply list entry for a ″C″ reply for message CPA4259 based on either the *job name* or the *user name* associated with the message.

  ```
  o  ADDRPYLE SEQNBR(56) MSGID(CPA4259) CMPDTA(QPADEV0011 27) RPY(C)

  o  ADDRPYLE SEQNBR(57) MSGID(CPA4259) CMPDTA(USRIDNAM   51) RPY(C)

  do not forget...   CHGJOB INQMSGRPY(*SYSRPL)
  ```

If you do not think the query selection information results in acceptable performance, additional detailed analysis of the query select statements and the existing database indexes is required.

Use query support when it is necessary to provide the application requirements. Native database access for simple record retrievals almost always can perform faster than queries because all queries go through a ″query component″ that native database interfaces do not.

Key SQL considerations include ensuring the Open Data Path (ODP) is kept open across invocations, an existing access path (index) is used rather than creating a new one, and using subqueries where possible. Subqueries allow multiple SQL statements to be treated as a single subquery operation.

12.6.2 Query Data Access Methods

Getting the query component to select the ″best″ data access method for your specific data organization and the number of records that are actually retrieved is important and has the following general considerations as shown in Table 23 on page 270.

Table 23. Summary of Query Data Management Access Methods

Access Method	Selection Process	Good When	Not Good When	Selected When	Advantages
Dynamic processing	Reads all records. Selection criteria applied to data in data space.	> 20% records selected	< 20% records selected	No ordering, grouping, or joining and > 20% records selected.	Minimizes page I/O through pre-fetching.
Key selection	Selection criteria applied to index	Ordering, grouping, joining	Large number of records selected	Index is required and cannot use key positioning method.	Data space accessed only for records matching key selection criteria.
Key positioning	Selection criteria applied to range of index entries. Commonly used option.	< 20% records are selected	> 20% records are selected	Selection fields match left-most keys select < 20% records.	Index and data space accessed only for records matching selection criteria.
Index from index	Key row positioning on permanent index. Builds temporary index over selected index entries.	Ordering, grouping and join operations	> 20% records are selected	No existing index to satisfy ordering but existing index does satisfy selection and selects < 20% records.	Index and data space accessed only for records matching selection criteria.

12.6.3 General Query Tips and Techniques

This section lists several query tips and techniques. Refer to the publications listed at the beginning of Section 12.5, "Queries and Structured Query Language (SQL)" on page 262 for additional details.

SQL and Reusable Open Data Paths (ODPs): Transaction (interactive) oriented applications that enable the already built ODP to be reused for each transaction deliver better performance as compared to having the system build a new ODP for each transaction. This is analogous to keeping files open across program calls for each transaction.

The CLOSE SQL CURSOR defaults to *ENDPGM in embedded SQL. In the transaction environment, any reusable ODPs is closed when the program ends. It is recommended that a higher level main program be designed to call lower level programs compiled with CLOSQLCSR(*ENDJOB) value to enable cursors to be opened once for the lifetime of the job.

You must place the job in debug mode and review messages to determine how the ODPs are being used.

Build Indexes Specific to the Queries:

Where high performance is required:

- Build the indexes to be consistent with the ordering and selection criteria. Use the optimizer messages to help determine the keys needed for best performance.
- Avoid putting frequently updated fields (keys) in the index.
- While creating the ″correct indexes,″ avoid creating unnecessary indexes and delete indexes no longer used.

  This reduces the time the optimizer takes to determine a good candidate index.

You should understand the optimizer″s ″key range estimate″ as described in the documents referenced in Section 12.5, “Queries and Structured Query Language (SQL)” on page 262. The key range estimate is a process the optimizer uses to estimate the number of records/rows that are selected based on the individual predicate. In the following example, this is ″how many records match X=10″. These estimates can only be performed precisely if an index exists with that value as the primary key.

```
SELECT * FROM FILE12
         WHERE X=10 AND Y=5
```

If there is an index with a first key of X or Y, the optimizer can perform a key range estimate on that predicate. If there is no primary index for a predicate, the optimizer uses a default filter factor (10% for an equal test).

Many applications do not have indexes built to contain statistics to assist optimization for predicates that are highly selective. Any index built in V2R3 or later contains internal statistics that assist the optimizer.

Many queries also build useless selection lists, typically based on unplanned input from an operator. These queries sometimes include selection tests for all fields but put in values that select all records for the individual predicate.

For example:

```
SELECT * FROM FILE1
         WHERE X BETWEEN 0 AND 999
```

If X is a zoned decimal number of size 3, this test selects all records/rows. However, without a primary key index, the optimizer guesses that only 25% of the records/rows are selected. The best solution in this case is to not build this useless predicate. Alternatively, ensure there is a primary key index on X so the optimizer can determine that it selects 100% of the records/rows.

IMPORTANT

Customers with large databases (over one million records/rows) need to be especially knowledgeable of key range estimates. The key range estimate process can take seconds to minutes to complete when the index size is quite large. During the estimate process, the data spaces are ″seized with share,″ which prevents any update to the file/table.

Specify Ordering Criteria on Left-Most Keys: Specifying ordering criteria on the left-most keys of the index encourages index use by the optimizer when arrival sequence is selected.

Consider ALWCPYDTA(*OPTIMIZE) when Ordering: When ordering over a field, experiment with the OPNQYRF and CRTSQLxxx command parameter ALWCPYDTA(*OPTIMIZE). This enables the query optimizer to consider using a **sort** of the records rather than creating a temporary index. The time required to retrieve the first row/record may increase because of the preceding sort, but retrieving the remaining records may be fast. Note that it is important that an index already exist to satisfy the selection criteria as this assists the optimizer in its estimating process.

Consider ALWCPYDTA(*OPTIMIZE) for Key Selection: If the key selection method has been chosen (job log message) because ordering was specified on OPNQRYF, consider using ALWCPYDTA(*OPTIMIZE) and COMMIT options as follows:

- ALWCPYDTA(*OPTIMIZE) and COMMIT(*NO) when commitment control is not being used.
- ALWCPYDTA(*OPTIMIZE) and COMMIT(*YES) when commitment control has been started with *CHG or *CS.

These options enable ordering to be performed with the query sort routing.

%WLDCRD (Wild Card) Predicate Optimization Consideration: When using the %WLDCRD predicate optimization, avoid using the wildcard in the first position.

A wildcard in other than the first position may assist the optimizer in selecting an existing index.

For Join Optimizations Minimize Secondary File Records Processed: It is important to minimize the number of secondary files/tables records/rows (″dials″ in V3R1 documentation) that are processed before finding each match. The following tips can help minimize this processing:

- Ensure indexes exist to supply join statistics to the system.

  For example, assume the following join SELECT statement:

  ```
  SELECT * FROM FILE1 A, FILE3 B
           WHERE A.ZIP = B.ZIP and A.SECTION = B.SECTION
  ```

  Indexes over A and B with leading keys of ZIP and SECTION (order does not matter) allows the optimizer to gather more accurate statistics on the optimal join order and index usage.

- Avoid ordering data from more than one file. Generally, this forces a temporary result file to be generated, which involves more overhead in achieving the query results.
- Avoid joining two files without a JFLD or QRYSLT clause. This includes not specifying JDFTVAL(*YES) on the OPNQRYF command, which may result in a large number of additional records being processed.
- Create an index on each secondary file to match the join fields.
- Ensure the fields used for joining files match exactly in field length and data type (attribute).
- Allow the primary file to be the file with the fewest number of records that are selected. This can significantly minimize the number of disk I/O operations necessary to find matching records/rows.
- When specifying ordering on more than one of the join files, experiment with using ALWCPYDTA(*OPTIMIZE) as discussed previously for ordering.
- Experiment with OPNQRYF OPTIMIZE(*FIRSTIO) or OPTIMIZE(*ALLIO). If the query is creating a temporary index and you feel using an existing index may improve performance, try specifying OPTIMIZE(*FIRSTIO).

  If the query is not creating a temporary index and you feel creation of a temporary index may improve performance, try specifying OPTIMIZE(*ALLIO). *ALLIO typically delivers good performance when the query retrieves almost all of the records available.

- Consider OPTIMIZE(*MINWAIT). In general, OPTIMIZE(*MINWAIT) can bias the optimizer towards building indexes. This frequently results in good performance when OPNQRYF is used to build an Open Data Path for a subsequent Higher Level Language (HLL) program to use.
- Consider SQL OPTIMIZE FOR n ROWS. SQL OPTIMIZE FOR n ROWS can bias the optimizer to select an existing index when smaller rows (n) are specified.

Avoid Numeric Data Conversions: In general, always use the same data type for fields and literals used in a comparison:

- Same data type
- Same scale, if applicable
- Same precision, if applicable:

  The optimizer may not use an index if these are different.

Avoid Arithmetic Expression Comparison: Avoid an arithmetic expression as the operand to be compared to a field in a record selection predicate. If you use the arithmetic expression, the optimizer does not use an index on the field being compared to the arithmetic expression.

Consider Deleted Records Impact: If arrival sequence is used frequently for queries and there is a possibility of many deleted records, use the Reorganize Physical File Member (RGZPGM) command or file REUSEDLT(*YES) (re-use deleted records) to remove or minimize the space occupied by deleted records.

Consider QRYSLT on OPNQRYF: If specifying selection expressions on the OPNQRYF command, use QRYSLT (Query Select Expression) rather than GRPSLT (Group Select Expression), if possible. This minimizes the selection processing overhead.

Consider using CHGQRYA DEGREE(*ANY): This causes the optimizer to *consider* using data pre-fetch tasks for each disk that contains all or a portion (extent) of the file/table data to be processed by the query. These parallel tasks bring data into main storage significantly ahead of when the data is processed by the query component, thus minimizing actual disk I/Os.

This pre-fetch processing is most effective for long running I/O-bound data space and data space index scan queries.

In addition to specifying DEGREE(*ANY), the query job must be running in a shared storage pool with expert cache (*CALC) enabled for the pool.

Each query must have main storage space available of at least 2MB to 4MB because it takes up to 1MB per input stream task. If the disk space extent is less than 1MB, less space is used by the task for that disk drive.

As a general guideline, you should assume 16MB per active query using this support.

Note: Since this parallel pre-fetch processing aggressively utilizes main store and disk I/O resources, the number of queries that use this method should be limited and controlled. More information pertaining to parallel pre-fetch can be found in *DB2 for OS/400 SQL Programming V3R6*, SC41-4611.

Parallel pre-fetch does not get used when building indexes. Maximum benefit is obtained when performing scan I/O requests such as:

```
SELECT * from SOMELIB/SOMETAB
        where COL1 = ′A′
```

Use SQL ″FETCH FOR n″ Records: In order for SQL implementation to retrieve blocks of records/rows under commitment control *CS and *ALL, your program must specify a number of records greater than one in the ″FETCH FOR n″ phrase.

A brief description of *CS and *ALL are included here for easy reference.

***CS** (cursor stability) means all records/rows that are read, updated, deleted, or added/inserted within a commitment control boundary are locked until a COMMIT or ROLLBACK operation is issued or a ″unit of work″ is ended. A record/row selected but not updated or deleted can individually be released if another record within the file/table is selected or an explicit unlock operation is issued.

***ALL** means all records/rows that are read, updated, deleted, or added/inserted within a commitment control boundary are locked until a COMMIT or ROLLBACK operation is issued or a ″unit of work″ is ended. A record/row selected but not updated or deleted cannot be individually released during the commitment control boundary.

Recreate SQL Applications That Come from Releases Prior to V3R1: Recreate all SQL applications that originated in releases prior to V3R1 to reduce intermediate internal calls to the QSQROUTE and other SQL run-time routines.

Consider Using an SQL Stored Procedure: Beginning with V3R1 and available in V3R6, SQL/400 supports stored procedures. A stored procedure is basically an already created program that is accessed either locally or remotely through SQL syntax. While a stored procedure may be used locally, it is more typically thought of as an efficient way to access a remote database while preserving the use of SQL within the ″calling″ (local) program.

In the remote access environment, a stored procedure enables the application developer to distribute the logic between a client and a server system. Also, it allows for a number of operations to be packaged in one request that may lead to lower traffic over the communications link between local (client) and remote (server) systems.

Since the stored procedure may be written in a high-level language (HLL) such as COBOL, RPG, or in a REXX procedure, the stored procedure may perform complex computations and access database data through either SQL syntax or AS/400 native file I/O interfaces. Figure 47 on page 275 shows an example of the SQL syntax that ″calls″ the stored program/procedure.

```
    EXEC SQL
     DECLARE ORDPRC PROCEDURE
        (:ORDNO IN DECIMAL(5), :TOTAMT DECIMAL(5,2))
        (EXTERNAL NAME MYLIB/ORDENT1
         LANGUAGE RPG)
    END-EXEC.
       :
       :
    EXEC SQL
     CALL ORDPRC (:ORDNO,:TOTAMT)
    END-EXEC.
       :
       :
```

Figure 47. SQL/400 Stored Procedure Example

Figure 47 shows the SQL CALL statement for the stored procedure (ORDPRC) that passes values in parameter ORDNO and parameter TOTAMT each time the procedure is called.

The DECLARE PROCEDURE statement is an optional statement that identifies the number and type of parameters that is passed on a subsequent call, the external name of the program (the name on the AS/400 system), and the language of the program or procedure on the remote system (″server″). Note that the parameters are converted as necessary, including converting data types, if required, for different languages. Using the optional DECLARE PROCEDURE statement may improve performance of the application because it reduces some of the work that is ordinarily done at the time of the procedure CALL. The CALL statement simply calls the procedure and passes the necessary parameters.

The stored program/procedure merely defines ORDNO and TOTAMT as parameters passed to it in the syntax of the stored procedure programming language.

In the preceding example, the DRDA CONNECT TO xxxx statement is required for the stored procedure to be run locally or remotely.

Remote Database Access Performance Considerations: Regardless of whether Distributed Data Management (DDM) native database file I/O interfaces or Distributed Relational Database (DRDB) SQL interfaces are used to access remote databases, they are not as fast as accessing the data on the local system.

In designing a remote data access application, the primary consideration is to have as much data on the local system. For example, if the master customer records are rarely updated, placing a copy on the local system eliminates the remote access to that data.

Blocking I/O, stored procedures, and (for Client Access/400 APIs) using ″combined operations″, can minimize the difference between local and remote access performance.

See other information in this subject area under the *client* and *server* index entries.

System-wide Catalog Performance: For V3R6, the system catalog information is stored in the following files in QSYS:

- QADBXREF (file xref)
- QADBPKG (SQL package info... includes Consistency Tokens)
- QADBFDEP (file dependency)
- QADBXRDBD (remote databases)
- QADBFCST (file level constraint xref)
- QADBCCST (constraints field usage)
- QADBIFLD (field xref)
- QADBKFLD (key field xref)

AS/400 database management provides views over these files, enabling more consistency with catalog views of other IBM SQL products, and with ANSI and ISO standards (called Information Schema). The views are found in the QSYS2 library.

Tables and views in the catalog are the same as any other database tables and views, and can be accessed by standard SQL statements.

The database manager ensures that the catalog is accurate at all times since it is constantly maintained.

In some cases, SQL SELECT performance on this catalog information was slower than using SELECTS directly on the internal files within library QSYS. This overhead may be due to data conversion requirements to preserve ANSI and ISO compliance.

If this compliance is not a consideration with the customer, consider using SELECTS directly on the internal files and creating logical views of these files to improve query performance.

12.7 Compile Time Performance

The purpose of this section is to provide you with general information on compile time performance when moving to V3R6. First, here is a list of changes in V3R6 that significantly impact compile time. The list begins with the two factors mentioned in the preceding section:

- RISC code generation
- OPM program template conversion
- Memory requirements
- Changes to ILE *BASIC optimizations
- The new ILE/C optimization level of 40

The following tables depict the relative performance of OPM and ILE compile time on V3R6 versus V3R1 for both RPG and COBOL. The V3R1 OPM compile time is used as a base (equal to 1.0). This information cannot be used to compare RPG to COBOL compile time performance.

The relative performance numbers included in the tables are averages based on results in a controlled environment where compiles were done at the lowest optimization level and sufficient memory was available to keep paging to a minimum. Depending on your compile workload, your results may vary.

*Table 24. V3R6 versus V3R1 Compile Time, Optimization *NONE*

Program Type	**RPG and COBOL**	
	V3R1	**V3R6**
OPM	1.0	1.2-1.7
ILE	Greater than 2.0	1.0
Note: A larger value signifies longer compile time.		

*Table 25. V3R6 versus V3R1 Compile Time, Optimization *FULL*

Program Type	**RPG and COBOL**	
	V3R1	**V3R6**
OPM OPTIMIZE(*YES)	1.0	4.4
ILE OPTIMIZE(*FULL)	4	3
Note: A larger value signifies longer compile time.		

The following table shows the relative performance of ILE C compile time on V3R6 versus V3R1. The V3R1 ILE C compile time is used as a base (equal to 1.0). As stated previously, the results were obtained in a controlled environment and your results may vary.

*Table 26. V3R6 versus V3R1 ILE C Compile Time, Optimization *NONE*

Program Type	**ILE C**	
	V3R1	**V3R6**
ILE C	1.0	0.5-0.8
Note: A larger value signifies longer compile time.		

12.7.1 Compile Time Conclusions

The V3R6 compile process requires more memory in V3R6 than V3R1. The following conclusions are based on using sufficient memory to keep paging to a minimum. Compile times are also sensitive to the optimization level.

- The compile time of both ILE RPG and ILE COBOL have improved significantly on V3R6. In both cases, V3R6 ILE compiles are, in general, up to two times faster than on V3R1 when compiling with optimization level (*NONE). The amount of improvement depends on the size of the program with larger programs realizing the most improvement.
- V3R6 ILE RPG and COBOL compile times for optimization level (*NONE) are on the average equivalent to OPM compile times on V3R1, but can range from about 20% slower up to 20% faster.
- ILE C compile time on V3R6 has improved significantly over V3R1 (on the average from 20% up to two times faster). The most improvement is in large programs at optimization level (*NONE). The least improvement is in small programs at higher optimization levels.
- V3R6 OPM compiles are approximately 20% to 70% slower than on V3R1 for non-optimized programs. The additional time is the result of the automatic translation to allow OPM languages to utilize the Optimizing translator.

- Optimized ILE RPG and COBOL compiles have improved 30% to 40%. OPM optimized compiles are much longer than on V3R1. This is due to the more difficult optimizations attempted by the V3R6 optimizing translator when compared with the V3R1 OPM optimizing translator.

12.7.2 Compile Time Recommendations

- When possible, application developers should move to ILE. In addition to improved compile times, ILE offers many advantages over OPM, such as modularity, static binding, common run-time services, and improved code optimization.

  ILE RPG is shipped with a command, CVTRPGSRC, that can be used to migrate your RPG III source code to RPG IV. Appendix B in the *ILE RPG/400 Programmer′s Guide*, SC09-2074, contains a detailed description of the conversion process with examples to help you identify and quickly resolve potential conversion problems. Another source for information on the conversion process is the redbook *Moving to ILE RPG*, GG24-4358.

  For conversion and compatibility considerations between OPM COBOL and ILE COBOL for OS/400, please refer to Appendix G in the *ILE COBOL/400 Programmers Guide*, SC09-2072.

- The following suggestions help in managing and improving compile times:

  – For initial compiles, use OPTION(*NOGEN), and optimization *NONE or *NOOPTIMIZE:

    *NOGEN compiles the module or program but does not create a program object. It can be used to fix and edit compile errors.

    Using optimization *NONE or *NOOPTIMIZE can dramatically reduce compile times. Optimized compiles can be expected to take at least three to five times longer than compiles at optimization *NONE or *NOOPTIMIZE. Once the application is debugged and ready for production use, compile it at the appropriate optimization level and conduct a final test. Typically, RPG and COBOL programs should be compiled at optimization level (*NONE) or (*BASIC) and C programs at optimization level (*FULL) or level 40.

  – Make sure that enough memory is allocated when compiling. This depends on the size of the program being compiled and the level of optimization.

  – Compile in batch rather than interactively.

  – The following recommendations hold for ILE applications:

    - Design modular applications:

      Modular programming offers faster application development and a better ability to reuse code. Programs are developed in smaller, more self-contained procedures. These procedures are coded as separate functions and bound together to build an application. By building applications that combine smaller and less complex components, you can compile and maintain your programs faster and easier.

    - Use the value of DBGVIEW adequate for your purpose:

      Requesting debug information requires more compile time and creates larger objects. For example, DBGVIEW(*LIST) results in a slower compilation time than DBGVIEW(*STMT). If the level of debug information you need is that provided by DBGVIEW(*STMT), selecting *LIST unnecessarily slows down compilation time and inflates the object size.

12.8 Display Workstation Programming Tips

Efficient use of workstation data management functions can dramatically impact response time results. This section includes workstation data management functions and performance tips that are also described in *Application Display Programming V3R6*, SC41-4715. See Section 12.2, "General Application Programming Tips" on page 230 for information on minimizing file open and close operations. Key considerations include:

- Minimizing the amount of data sent to or received from the display.
- Minimizing the amount of workstation specific data processing during a transaction.
- Minimizing the number of ″line turnarounds″ between the system and the display.

Note: Communication configuration parameters also affect performance. These parameters include frame size, SNA RU size, SNA pacing values,and token-ring acknowledgement frequency. *OS/400 Communications Management V3R1*, SC41-3406, provides detailed information on line protocol performance. The *Performance Capabilities Reference for PowerPC Technology*, ZC41-0607, provides laboratory test results considering these parameters in various environments.

11.12.2, "Communication Line Speed Considerations" on page 212 has general configuration parameter considerations.

12.8.1 Minimizing Data Sent or Received

Display File RSTDSP(*YES) Parameter: If the currently open display file has RSTDSP(*YES) and a program issues a full open to a display file, an image of the current display and device status is ″saved″ and sent to the system to be used in a later ″restore″ function. This image is ″restored″ when the program with the previously open display file regains control and issues a write or read operation to that file. These display save and restore functions are often used to ease application development of individual programs without consideration of the order of program invocation.

However, frequent use of save and restore display functions can cause up to 2000 bytes of data to be transferred. Although the 5294 and 5394 microcode Release 2.2 provides save/restore data compression, full displays of data may still require a significant amount of data transfer. The more frequently a save/restore display is done and the slower the ″line speed″, the more dramatic the performance degradation.

For maximum performance, both programs should be designed to share the same display file (SHARE(*YES)) or each program must be responsible for re-formatting the display when control is returned to it from the CALLED program. RSTDSP(*NO) should be used.

Using a shared display file also significantly reduces general file open processing overhead. Note, however, that if a shared display file contains a large number of record formats (such as 50), the Process Access Group (PAG) can become excessively large. This significantly degrades system performance if a large number of jobs use this display file on a system with constrained main storage, even if a specific job uses only a few of the record formats.

If a program uses certain functions, such as Put Override (PUTOVR) or Clear Lines (CLRL), and calls another program that does not use a shared file, either RSTDSP(*YES) must be specified or when control is returned to the calling program, the program is responsible for complete re-formatting of the display.

Minimize the Use of Break Messages during Peak Workloads: Break message support must save and restore the display independent of the RSTDSP parameter of the currently opened display file.

The DDS Windows Support (WINDOW, WDWBORDER, and so on): This support enables easier development of window overlays. However, displaying a new window does, in most cases, require workstation data management to use the 5250 read screen and save and restore functions. As the frequency of read screen and save/restore operations increase, line utilization increases. This can cause performance problems on slow speed or heavily utilized communication lines.

The DDS keyword USRRSTDSP (user handles restoring displays) can be used to bypass the save and restore functions. This support requires the user program to be responsible for showing the display that was overlaid by windows. See *OS/400 Application Display Programming V3R6*, SC41-4715, for additional details.

General Display I/O Considerations

- For output operations, minimize the number of fields sent to the display. Combine constant data into a single constant field wherever possible. Keep the number of input capable fields to a minimum as each time a new input field (not currently on the display) is sent, additional 5250 data stream characters are required.
- For repeated output operations for the same record format (for example, second through nth order dialogue), use Put Override support (PUTOVR, OVRDTA, OVRATR) that specifies only the field or field attributes that need to be changed. For example, do not re-send constant fields or customer number.
- For repeated output operations, also consider using Clear Line No support. You can use the Clear Line function (DDS CLRL(*NO)) when existing display data is being updated (overlaid) with new data. CLRL(*NO) can be used in cases where changed data overlays constant (prompt) data. For example, FORMAT1 contains only constants and FORMAT2 CLRL(*NO) contains only changed data.

  Using CLRL(*NO) and Put Override provide some similar functions. Experimentation may be required to determine what is most efficient versus ease of programming for a particular application.
- For repeated input operations, use DDS ERASEINP (erase input) on the intervening output operation to blank out input capable fields to get ready for the next input operation.

  ERASEINP causes a 5250 command to be sent to the device rather than sending out blank characters for each field to be cleared. Use INZINP (initialize input fields) with ERASEINP and PUTOVR to initialize the input field save area without sending blanks to the device.

- For output operations, minimize the number of different record formats on the display at the same time. If multiple formats are required and more than one of them contains input capable fields, the formats should be sent in ″top-down screen order″. For example, assume FORMAT1 has input fields in line 2 through line 4 and FORMAT2 has input fields in line 6 through line 10. Outputting FORMAT 1 first and then FORMAT2 is more efficient than a FORMAT2, FORMAT1 sequence. (The 5250 hardware requires input field definitions to be received in ascending row/column order.)

  Within the same format, CRTDSPF support orders the fields in ascending sequence. But programs using User-Defined Data Stream (UDDS) support have complete data stream responsibility.

- Ensure the primary display size (DDS DSPSIZ) is used on most workstations. The primary display size is defaulted to 24 by 80 or is specified as the first value in the DSPSIZ keyword. Additional system processing is performed when the display device being used is not the primary display size.

- Use Attention keys rather than Function keys. Through DDS CFnn or CAnn, a 5250 command key can be defined to return data (CF - function) or just return the Attention ID (AID) key and cursor location (CA - attention). When only indication of the key is necessary for the application, define the key as CAnn.

- For output operations, minimize editing output fields (DDS EDTCDE, EDTWRD). Editing a few fields is acceptable but can consume excessive CPU as the number of edited fields increases.

12.8.2 Minimizing the Amount of Workstation Specific Data Processing

There are two major areas of processing display specific data: validity checking input data, and using AS/400 display subfile support.

12.8.2.1 Data Validity Checking

Minimize the Use of 5250 ″Numeric Only″ Field Types: In DDS, these are defined as ″Y″ type fields and permit the operator to enter formatting characters such as ″$″ and decimal indications (″.″ or ″,″). On input, workstation data management must scan for these characters, perform decimal alignment, and remove these characters for the field returned to the program. If there are a large number of these field types processed by each transaction, excessive CPU can be consumed when many jobs are using this facility.

Use DDS Keywords: Use DDS keywords to remove as much data validation and error notification processing as possible from the application program.

Use DDS keywords such as CHECK, COMPare, and RANGE to get data management or the device itself to do as much validity checking as possible. When the device can do the checking, no data is sent to the system. When data management does the checking, minimal data is exchanged between the system and the device, and data management code is more efficient than using high-level language instructions.

When the user program has to perform some validity checking, use the DDS ″error message keywords″ to minimize error indication data sent to the display. Error message keywords include ERRMSG, ERRMSGID, MSGID, and MSGCON.

12.8.2.2 Subfile Support

A subfile can contain many records. If a program processes a large number of those records per transaction, response time can be poor. The objective is to process as few changed (input) or new (output) records as possible and display the appropriate page of the subfile even if there are more records to process.

Minimize the Number of Subfile Records Written to the Subfile: A subfile has two ″size″ parameters (subfile size and page (display) size). As soon as a page worth of records has been added or updated, the page should be sent to the display, rather than first writing all new or changed records to the subfile and displaying a page.

Programming this support can be fairly simple or can be made more complex.

In the simple case, the program can complete a page of subfile records and display that page (write and read to the subfile control record with SFLDSP and SFLCTLDSP keywords in effect and SFLRCDNBR controlling the page to be displayed). With use of the ROLLUP/PAGEDOWN keyword with a response indicator, control is returned to the program when the operator uses the Rollup/Pagedown key. The program can build the second page of the subfile and cause it to be displayed. Now workstation data management can display the first and second pages of the subfile without returning control to the user program.

When the Rollup/Pagedown key causes data management to display a subfile page that it does not yet have, the Rollup/Pagedown indication is returned to the program so that a subsequent page can be built and displayed just as for the second page. This support makes use of subfile support and minimizes response time.

User programming can be made more complex but with improved subfile display response time if the write and read issued to display the page is a write with DDS INVITE followed by read-filename sequence. The write with INVITE operation returns control to the user program while the operator is viewing the display data. While the operator reviews the data, the program can be actively building one or more additional pages within the subfile. When ready, the program can issue the read-filename operation and process any input from the operator. The read-filename operation invokes the data management ″read-from-invited-devices″ operation.

If the normal operator interaction with the application is to display the second page of the subfile after viewing the first, the response time for displaying page two is minimal with this technique. This programming technique should be used only in a well-understood environment, since building of the subfile keeps the job in the activity level longer.

Minimize the Number of Subfile Records Processed on Input: With a large subfile, it is possible that an operator can enter several input records. Even when the subfile operation ″read-next-changed″ is used, the end user may have typed many data records. If a large number of input records are processed by the program, a long response time can result when the ″updated″ subfile is displayed.

Processing all of the changed (input) subfile records before displaying a subfile page should be compared against the total number of transactions processed rather than response time. In other words, processing a single line item of an order without subfile support gives good response time per line item. Processing all line items before writing to the display gives noticeably poorer individual response time. However, processing all of the subfile records may result in more orders per hour than the ″line item at a time method″.

Choosing to process all subfile records rather than a line at a time may require some end user re-training.

Subfile Initialization Considerations: When a particular sequence of subfile operations has been completed, such as for an order completion, consider using the SFLCLR (clear subfile) and SLFINZ (initialize the subfile) keywords rather than multiple user program output operations to clear the subfile of old records.

12.8.3 Minimizing Line Turnarounds

Line turnarounds are, in general, communication protocol type-dependent functions that indirectly affect performance. When they can be avoided or at least minimized, response time is improved. While some AS/400 ″device type″ support gives the user programmer explicit control over line turnarounds (such as RETAIL support) and communication configuration parameters can influence the frequency of line turnarounds, the workstation programmer can implicitly perform programming functions that affect line turnarounds without realizing it.

Using display file RSTDSP(*YES) and full file opens increase the number of line turnarounds. For workstation programs, using ″output only″ operations cause a communication line turnaround that is transparent to the user program but has a dramatic affect on response time based on the number of line turnarounds and communication line speeds.

Line turnarounds often cause AS/400 jobs to remain in the activity level waiting for an SNA response. In SNA, this is called ″Definite Response Mode″. On the AS/400 system, this waiting is called a ″short wait″ in the Performance Tools reports. If the short wait lasts longer than two seconds, this becomes a ″short wait extended″. If the short wait becomes extended, the job is taken out of the activity level. When the SNA response is received from the device, the job is ready to run again.

The following list contains programming considerations for minimizing workstation line turnarounds:

- Many RPG programs use the execute format (EXFMT) operation code to write and read (″put-then-get″) to a display file. Using this operation code eliminates a line turnaround until data is received or an error occurs.
- An output only operation, such as an RPG/COBOL WRITE statement, causes a line turnaround unless the display file being used either has a file with DFRWRT(*YES) specified, or the DDS keyword INVITE is used on the format identified on the output operation. Using DFRWRT(*YES) typically resolves the occurrence of a line turnaround by holding the output data until an input operation is issued by the program.

  If the INVITE keyword is used, the high-level language programmer must understand multiple device file support according to the syntax of that language. The program must issue a read operation to receive data or detect an error condition. The ″read-filename operation″ can make use of a timeout (display file WAITRCD parameter) when waiting for input after an INVITE operation. The high-level language read-filename invokes the data management read-from-invited devices operation. The program invocation stack shows module QDMACCIN if read-from-invited-devices is used.

The Write with Invite followed by Read sequence enables the program to do additional processing before issuing the read operation to time out on the read operation, and to process input data from display files, ICF files, and data queues. These capabilities are discussed in various topics in this section.

Note: The ″Write with Invite followed by Read″ sequence does consume more CPU than the typical ″put-then-get″ data management operation used in most workstation applications. RPG′s EXFMT operation and CL′s SNDRCVF command use the ″put-then-get″ operation. Equivalent COBOL support through the WRITE and READ statements requires DFRWRT(*YES) on the opened display file. Only use the Write with Invite followed by Read sequence if you need the functions available through this interface.

DFRWRT(*YES) is the system default for display files, but the program with a sequence of several write only operations followed by a read must be aware that no output data is displayed until one of the following operations is issued by the program:

- Read operation
- Write and Read (Put then Get) operation
- Write format with DDS INVITE operation
- FRCDTA DDS keyword is in effect for a write only operation.

- Some applications have the need to update a display without requiring operator input. Examples are updates to stock status or delivery truck status displays where operator input is optional, but status changes frequently. (Other jobs update the status.) In this case, the best performance is achieved when no input operation, including DDS Invite, are outstanding to the device.
- Consider use of the ″Write-with-INVITE and Read-filename″ support when needing a ″READ time out function″ or when input data is either from a workstation or data queue.

  The Read-filename can use the display file WAITRCD timeout support to return control to the program. The program can examine any status change on a database file, message queue, or data queue. If there is a change, the program can issue another Write-with-INVITE that updates the display. The output format must either use Put Override support or the OVERLAY keyword to ensure any data currently being typed by the operator is not discarded.

  The program must process data that may have been received by the system as the write operation was sent. Major/minor return code 0412 indicates if data has already been received. You should consider using the DDS keyword Retain (keyboard) Lock Status (RETLCKSTS) to reduce loss of data being typed when the display is updated.
- Some programs need to be able to process input data from either a workstation or a data queue. This can be done if a data queue is specified on the display file through the DTAQ parameter. The program must examine the ″read-filename operation″ completed information to determine what kind of data is ready to be processed. Data queue data must be retrieved through a call to the QRCVDTAQ program. This technique minimizes unproductive examinations for updated status by the program that reduces CPU usage.

- Minimize consecutive Write-with INVITE operations when using multiple consecutive writes to the display.

  In some application environments, especially migrated System/36 applications, the program does several writes to the display before being able to process the incoming data. Often these programs use the INVITE function (DDS keyword or migrated SFGR format) on each output operation. The INVITE function gives the workstation ownership of the capability to send data. A subsequent write operation causes the system to perform an SNA dialogue to regain the ability to send data. This ″cancel invite″ scenario requires extra data transmission and time delays that can become noticeable on highly utilized remote communication lines.

  Though improvements have been made in speeding up the process of getting the system the capability to transmit data, it is much more efficient to turn off the INVITE on all but the last Write operation preceding the Read operation.

- Programs using User-Defined Data Stream Support (UDDS) have the same high-level language line turnaround considerations as programs using normal workstation data management support. In addition, the UDDS output record doing an RPG EXFMT or write with DDS INVITE operation must also specify a hexadecimal value of X′63′ or X′73′ (Send/Receive) in byte 5 of the output ″data″. X′73′ is for a 5250 display and X′63′ is for a 3270 remote display.

12.8.4 Designing the Primary Display Size for a Display File

Normally, the display files are set up for a 24 by 80 display (default size). The ***DSPSIZ*** keyword specifies which display sizes are valid for a file and indicates which sizes are the primary and secondary display sizes. (The primary display size is the first or only DSPSIZ value.) On the ***DSPSIZ*** keyword, the display size can be specified as ****DS3, *DS4, (24 80), or (27 132).*** For example, ***DSPSIZ (24 80)*** specifies a display size of 24 by 80.

The display size designated as the primary display size should be the one with which the display file is most often used. A performance benefit is realized by coding the ***DSPSIZ*** keyword in this manner. Additional processing is performed when the actual display size is the secondary display.

12.8.5 Sending Records with the Input Fields to the Display in Order

Records containing input fields should be sent to the display station in the order in which they appear on the display. This technique provides better performance than if record formats with input fields are sent randomly or in some other order.

12.8.6 Restoring the Display

When ****YES*** is specified for the Restore Display ***(RSTDSP)*** parameter, an image of the current display is saved when the display is suspended. When the display file is activated again, the saved image is used to restore the display to its appearance before being suspended.

The ***RSTDSP(*YES)*** parameter must be specified for the following keywords. If the parameter is not specified, data on the display can be lost if the file is suspended.

- CLRL
- OVERLAY

- PUTOVR
- PUTRETAIN
- ERRMSG
- ERRMSGID

If none of the previous keywords are used, you can improve performance by specifying *NO for the RSTDSP parameter.

12.9 Choosing Batch File Transmission Techniques

Many customers have the need to exchange large files with other systems. The AS/400 system provides many different facilities for performing this file transfer. Each technique has its distinct user interface and performance considerations and the choice of one facility over another is often dependent on whether a customer is more familiar with one interface than another.

The *Performance Capabilities Reference for V3R2*, ZC41-8166, provides extensive information on many AS/400 batch file transfer facilities. In addition to performance considerations, the choice of selecting one technique over another depends on the end-user functions available with a specific capability and any previous experience the customer may already have in this area.

This section lists some of the software support available on the AS/400 system for transferring files available through Version 3 Release 6. ″(OS/400)″ on the item indicates that the support is included as part of base OS/400 support. In most cases, a complementary product is required on the other system:

- Third party application package
- (OS/400) user-written program using the Inter-System Communications File (ICF) interface. Typically, the program is written in a separately priced high-level language.
- (OS/400) user-written program using the SAA Common Programming Interface - Communications (CPI-C). Typically, the program is written in a separately priced high-level language.
- (OS/400) Send Network File function under the Object Distribution Facility (ODF) support
- (OS/400) Copy File using Distributed Data Management (DDM)
- (OS/400) Distributed System Node Executive (DSNX) support for files with the System/390 through the NetView Distribution Manager (NetView/DM) product
- SystemView System Manager/400 and Managed System Services/400 products for exchanging files with other systems including the System/390 through the NetView Distribution Manager, another AS/400 system, RISC System/600 with NetView/6000, and IBM personal computers with NetView/2
- AS/400 Remote Job Entry Facility and MVS/VM Bridge support available with the AS/400 Communications Utilities, program number 5716-CM1. This support is used for exchanging files with a System/390. Using RJEF may require a host program on the System/390 to reconcile record length differences between the AS/400 system and System/390 and the record transmission lengths used under RJE.

  System/390 RJE host support has been available for a number of years and is still chosen by many customers for file transfer.

- NetView File Transfer Program (FTP) program number **5730-082** for AS/400 communication. (MVS NetView FTP program number **5685-108** is required to exchange files with MVS.)
- (OS/400) Distributed Relational Database (DRDB) can be used to transfer data between the SAA platforms using SQL statements in a user-written program. This is especially useful if DB2 or SQL/DS must be accessed from an AS/400 system as it saves writing or purchasing a mainframe application to process the relational database. However, DRDB is more suitable for occasional remote table access than for batch data transfer.
- Client Access/400, program number 5716-Xx1: Two functions are provided for exchanging file data between the AS/400 system and the Personal Computer: File Transfer and Shared Folder support.
- Client Access/400 with appropriate client software (original DOS, Windows, OS/2 Clients, Client Access/400 for Windows clients, and Client Access/400 for OS/2 clients):

  This support also provides file transfer and network drive support similar to the Client Access/400 support.
- TCP/IP Connectivity Utilities/400, program number 5716-TC1, with both a user program interface and a workstation operator interactive interface to File Transfer Protocol (FTP)
- OSI File Services/400, program number 5716-FS1, with both a user program interface and a workstation operator interface to FTAM (File Transfer Access and Management) support
- AIX Viaduct for AS/400, program number **5730-078**. This support enables RISC System/6000 to perform SQL functions on the remote AS/400 database and return the results to the RISC System/6000 program.
- AIX AS/400 Connection Program/6000, program number 5621-051. This support provides 5250 emulation on the RISC System/6000 and file transfer support on both systems. Exchanging file support is similar to the capabilities available between the AS/400 system and the personal computer running AS/400 Personal Computer Support/400.

With all of these facilities available to choose from, it is sometimes difficult to choose one over the other, and in some environments, multiple facilities may be the best solution. Base performance considerations include:

- Assuming minimal data processing in the program, user-written programs on both the sending and receiving systems can deliver the best response time, but may require programming development time and exclusion of some ease of use facilities, such as re-sending the data after a communication line failure.
- CPI-C versus ICF interfaces for batch file transfer applications:

  CPI-C interfaces are slightly more efficient than ICF file interfaces. Better performance throughput is observed with large batch file transfer applications.

  With V3R6 APPC support, both RPG and C provide slightly more efficient interfaces to ICF file interfaces compared to CPI-C. This is enabled by further reducing the movement of application data between the application and APPC data management. This support is somewhat similar to the familiar database support for SEQONLY(*YES), where the High Level

Language program does the database record blocking and deblocking for sequential input or sequential output.

In order to take advantage of this more efficient V3R6 ICF interface, the application programmer must:

- Use MAXPGMDEV(1) on the ICF file (default).
- Specify SHARE(*NO) on the ICF file (default).
- Use an ICF file with a separate indicator area (DDS INDARA keyword).
- Compile the program on V3R6.

By reviewing trace job printed output, you can verify this new support is being used. Look for the ″Locate Mode″ identifier **1** next to the application data.

```
 5716SS1 V3R6M0  940909                         AS/400 TRACE JOB INFORMATION                    04/03/96 15:11:53        PAGE    1
  TRACE TYPE  - *ALL     MAX STORAGE- 12000      EXIT PROGRAM-            *NONE
  RECORD COUNT- 054559   START TIME - 15:09:22   START DATE  - 04/03/96                JOB- 092501    /QPGMR     /CSTITMSCN
                                                                                       DB      NON-DB   PAGES     NUMBER
TIME          SEQNBR  FUNCTION  PROGRAM     LIBRARY     ENTRY EXIT  CALL LVL CPU TIME  READS   READS    WRITTEN   WAITS
15:10:19.877  017927  DATA      1C          4100638000000000000048000000000000000000000000000000   * Ä       ç                *
15:10:19.878  017928  DATA      1C          4100400290008001140000000000000000000000000000000000   *   °                      *
15:10:19.880  017929  RETURN    CSTITMSRN   CMN38       0275  0274     03       0.003      0       0        0           0
15:10:19.883  017930  CALL      QICPUT      QSYS        0001  0D57      04      0.032      0       1        0           0
15:10:19.885  017931  DATA      1C          F1C3E2E3D9C3C440404040C3E2E3D9C3C4404040400000800000   *1CSTRCD    CSTRCD          *
15:10:19.886  017932  DATA      1C          F20063F0F0F0F0F0F0F0F0F0F0F0F0F0F0F00000000000000000   *2 Ä000000000000000         *
15:10:19.888  017933  DATA      1C          F580000000000000000077C01784000D07404040404040404040   *5         Ï{ d             *
15:10:19.890  017934  DATA      1C          1AF0F200000000F0F0F0F0011440F0F0F0F9F2F4C3E4E2E340D5   * 02    0000   000924CUST N *
15:10:19.891  017935  DATA      1C          1AC1D7D7C3F140404040400E03C3E2E3D9C3C4404040400063D5   * APPC1       CSTRCD     ÄN *
15:10:19.893  017936  DATA      1C          1A000000000000000000000000000000000000000000000000D5   *                         N *
15:10:19.894  017937  DATA      1C          1A00400290008000000000000000000000000000000000000000   *    °                      *
15:10:19.904  017938  DATA      1C          1AF0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0   * 0000000000000000000000000 *
15:10:19.906  017939  DATA      75          000000000E4F01140000D1D30040404040404040404040404040   *     |    JL               *
                                                                                                         1
15:10:19.907  017940  DATA      75          40C9C3C640D3D6C3C1E3C540D4D6C4C54B4B4B4B404040404040   * ICF LOCATE MODE....       *
15:10:19.909  017941  DATA      75          40C1C340C6D440D3D6C3C1E3C540D4D6C4C54B4B4B4B40404040   * AC FM LOCATE MODE....     *
15:10:19.911  017942  DATA      75          000000000D3701140000D1D30040404040404040404040404040   *          JL               *
15:10:19.912  017943  DATA      75          40C9C3C640D3D6C3C1E3C540D4D6C4C54B4B4B4B404040404040   * ICF LOCATE MODE....       *
15:10:19.914  017944  DATA      75          40C1C340C6D440D3D6C3C1E3C540D4D6C4C54B4B4B4B40404040   * AC FM LOCATE MODE....     *
15:10:19.915  017945  DATA      1C          41C3E2E3D9C3C44040404000000011400000000000114000000   *CSTRCD                    *
15:10:19.917  017946  DATA      1C          4100638000000000000048000000000000000000000000000000   * Ä       ç                *
15:10:19.920  017947  DATA      1C          4100400290008001140000000000000000000000000000000000   *   °                      *
15:10:19.922  017948  RETURN    CSTITMSRN   CMN38       0275  0274     03       0.002      0       0        0           0
15:10:19.924  017949  CALL      QICPUT      QSYS        0001  0D57      04      0.032      0       2        0           0
15:10:19.935  017950  DATA      1C          F1C3E2E3D9C3C440404040C3E2E3D9C3C4404040400000800000   *1CSTRCD    CSTRCD          *
```

Figure 48. APPC Trace Job Example

- Minimize OSI use to exchange large amounts of data. Its performance is significantly slower than V3R1 TCP/IP or APPC. Rather, use OSI when communicating with a non-IBM system that does not support standard AS/400 file transfer capabilities.
- Use APPC for best throughput when exchanging large amounts of data between AS/400 systems.

  AS/400 APPC is usually faster than TCP/IP for large data transfers.

  V3R6 APPC is faster than V3R6 TCP/IP, but in some cases batch file transfers are within 1% to 5% of each other. Because of the TCP/IP implementation on the AS/400 system, TCP/IP uses more CPU resource than APPC.

  See Section 11.12.11, “General V3R6 APPC and TCP/IP Performance Expectations” on page 226 for more information on V3R6 APPC and TCP/IP performance.

- Distributed Data Management file transfer through Copy File is competitive with other file transfer facilities. DDM offers relatively straightforward processing and minimal ″setup time″. However, DDM does not provide any retry capability.
- Send Network File (SNDNETF) support is efficient during data transmission and offers the advantage of time-of-day transmission scheduling and auto retransmission if a communication error occurs.

  The disadvantage of SNDNETF and associated SNADS support is that the participating systems need to be configured for SNADS and, if large files are being transferred, the copying of large amounts of send data into an internal space may ″tie up the job″ for minutes.
- Exchanging large amounts of data concurrently in both directions is the most important way to make use of full-duplex lines. Mixing concurrent batch and interactive data exchange over the same line is not recommended. See information under Communication Line Considerations for some ″tuning″ that **may** result in acceptable performance.

12.10 APPC Programming Tips and Techniques

This section contains useful information in how to maximize APPC performance and reduce communications overheads.

12.10.1 APPC Session Negotiation

Normal ″varying on″ an APPC device description requires dialogue overhead that includes negotiating the type and number of concurrent APPC sessions and conversations permitted based on the APPC Mode Description parameters (MAXSSN, MAXCNV, LCLCTLSSN, and so on). CPU utilization for this work is assigned to system job QLUS.

This processing also occurs when the Change Session Maximum (CHGSSNMAX) command is issued to temporarily modify the values specified in the mode description.

Typically, this overhead occurs only for short periods of time. However, if hundreds of APPC device descriptions are varied on or CHGSSNMAX commands are issued over a short period of time, high (10% to 25%) CPU utilization may be charged to QLUS.

There have been cases where user programs on the remote system have repetitively used the API equivalent to Change Session Maximum. While architecturally acceptable, this has caused high CPU utilization on the local AS/400 system for job QLUS and degraded overall APPC transaction performance.

Note: If you detect QLUS CPU utilization over 10% for several Performance Monitor time intervals, determine if a large number (for example, over 50) remote APPC programs are doing this ″session negotiation″ within a brief time period. This requires either a trace of the communication line, or some ″remote program examination″.

An application design for a large number of remote systems communicating with a single ″server″ AS/400 would need to consider minimizing starting and stopping APPC conversations and sessions for each ″transaction″ and not using

″change session maximum″ as part of the ending of a conversation. The recommended application design includes:

- The non-AS/400 remote APPC programmer should simply use the SNA ″Unbind″ verb when ending a conversation (not preceding the Unbind with negotiating the number of sessions to 0).

  This technique is used by IBM applications.

- Determine with the customer which remote connections need to maintain the APPC conversation connection across multiple remote system ″transactions″.

  In other words, keep the conversation active and a session that is already bound, still bound.

While this topic discusses how to end conversations and sessions efficiently, keeping the active conversation for frequent exchanges of small amounts of data is recommended where possible. Keeping the conversation up bypasses OS/400 job start overhead, even when using *pre-started jobs*.

12.10.2 Maintaining APPC Conversations

The AS/400 system has significant overhead when starting a job as a result of an incoming program start request (APPC Evoke). In performance, critical data exchanges do not end the conversation when a single transaction has completed. Leave the conversation active so that job startup overhead is bypassed on subsequent data exchanges. End the conversation when the system-to-system connection is no longer desired, such as when a remote PC is shutting down.

12.10.3 Using Pre-Started Job Support

OS/400 provides pre-started job support to speed up *incoming program start request processing* (job initiation processing on the AS/400 system). A pre-started job is most useful for short transaction connections where job startup overhead is a significant portion of the total connection time. A pre-start job is defined to a subsystem monitor through the Add Prestart Job Entry (ADDPJE) command and Change Prestart Job Entry (CHGPJE) command. These commands specify which programs to start and how many copies of the pre-started job to make active and maintain through the operating environment. The job′s user profile is also specified.

The pre-started jobs can be defined to start when the subsystem monitor job starts or later through the Start Prestart Jobs (STRPJ) command.

Standard OS/400 subsystem monitor support detects situations when a large number of pre-started jobs have been active followed by periods when few of these pre-started jobs are actually doing work. When this is detected, the subsystem monitor gradually reduces the number of non-busy pre-started jobs.

When properly written (for example, opening database files before doing an ″acquire/allocate session″ operation), there is a significant performance improvement in the initial connection with the remote system that sent the remote program start request to the AS/400 system compared to complete job startup on the AS/400 system.

In V3R6, some IBM applications such as the Client Access/400 Windows, OS/2 clients, and host server programs use pre-started jobs. Refer to the *OS/400*

Work Management Version 3, SC41-4306, for more details on pre-started job considerations.

12.10.4 Routing Program Starts to Your Subsystem

OS/400 subsystem monitor support provides the capability to route incoming program requests to your own subsystem. This can be used to ″partition″ certain communication jobs into a subsystem other than QCMN.

While you can use routing entry compare values in subsystem QCMN to have jobs run in a specific storage pool and with a specific priority, you may want to have your own subsystem with its own storage pool (or pools) and prestart job entries. To do this, you have to specify communication entries more ″specific″ than the DEVICE(*ALL) and MODE(*ANY) communication entry shipped with QCMN.

To have incoming program start requests routed to your own subsystem, perform the following steps:

1. Define your subsystem, storage pools, and class descriptions you want to use.
2. Add a communication entry that specifies a specific device name, remote location name, or device type (for example, *APPC). For an APPC connection, you may also create your unique mode description and specify this mode name instead of the device name, remote location name, or device type.

   Note that depending on the type of APPC security being used (″secured location″ yes or no), you need to determine if your communication entry has to specify a default user profile as well.
3. Add one or more routing entries that specify a compare value of ″PGMEVOKE″ in starting position 29. You may also specify a specific program name compare value starting in position 37 or a unique APPC mode name compare value starting in position 1. Each of these entries may be used to specify the desired class description and subsystem storage pool assignment.
4. If subsystem QCMN is already active, you may have to vary the remote control unit and device description off before proceeding to the next step.
5. Start your subsystem monitor.
6. If the remote control unit and device description are varied off, vary them on.
7. When the remote APPC device varies on and sends in a program start request (with the appropriate APPC mode if a mode is specified in the communication entry), the remote device is allocated to your subsystem.
8. Once you have verified proper allocation to your subsystem, you should put the ″STRSBS your-system″ command in the system Startup program, preceding starting QCMN subsystem.

Refer to *OS/400 Work Management Version 3*, SC41-4306, for additional information.

12.10.5 Minimize Using Confirm and Force Data Functions

APPC data management support provides the ″confirmation″ protocol between systems. This support enables the sending system to ask for a confirmation or receipt of data from the remote system. This support requires an SNA definite response from the remote system, but *does not require a line turnaround followed by application data that ″confirms″ the receipt of data on the originally receiving system.*

When the confirmation of receipt of data and no line turnaround processing is required, use the ″request confirmation-confirmed″ APPC verbs.

Note that using ″force data″, ″confirm″, V3R1 two-phase commit ″prepare to confirm″, and ″receive prepared confirm″ should be used only where absolutely necessary for the application. These functions do slow down a transaction and take extra CPU cycles when used excessively.

In DDS, the sending system uses the CONFIRM keyword and the receiving system uses the RSPCONFIRM keyword. With CPI-C, use the CMCFM and the CMCFMD (confirmed) verbs, respectively.

There are times when a cooperative processing application needs to send data immediately to a remote system without requiring confirmation or a line turn around. Use the ″force data″ function on the AS/400 system to send the contents of any output buffer immediately to the remote system.

In the APPC architecture, *flush data* is the term used.

In DDS, the sending system uses the FRCDTA keyword. With CPI-C, use the CMFLUS verb.

12.10.6 Recommendations for Sockets and the PASCAL API (TCP)

- TCP using the sockets interface performs approximately twice as fast as when the PASCAL API is used.
- As the sockets interface does not block up multiple application sends, it is important to limit the number of interactions.

12.10.7 Recommendations for AnyNet

- Having ANYNET = *YES causes extra CPU processing. Have it set to *YES if it is needed functionally; otherwise, leave it set to *NO.
- For send and receive pairs, the most efficient use of an interface is with its ″native″ protocol stack. That is, ICF and CPI-C perform the best with APPC, and Sockets performs best with TCP/IP. There is CPU time overhead when the ″cross-over″ is processed.
- Each interface/stack may perform differently depending on the scenario. The bottom of the APPC stack is slightly more than the TCP stack. ICF and CPI-C APIs provide buffering that reduces overall CPU time; sockets does not provide this level of buffering.

12.10.8 Recommendations for DDM, ODF, FTS

- Copyfile with DDM provides an efficient way to transfer files between AS/400 systems. DDM provides large blocking which limits the number of times the communications support is invoked. It also maximizes efficiencies with the database by doing fewer large I/O′s. Generally, a higher rate can be achieved with DDM compared with user-written APPC programs (doing database accesses) or with ODF.
- When ODF is used with the SNDNETF command, it must first copy the data to the distribution queue on the sending system. This activity is highly CPU intensive and takes a considerable amount of elapsed time. This time is dependent on the number and size of the records in the file. Sending an object to more than one target AS/400 system requires one copy to the distribution queue. Therefore, the realized data rate may appear higher for subsequent transfers.
- FTS is a less efficient way to transfer data. However, it offers built-in data compression for line speeds less than a given threshold. In some configurations, it compresses data when using LAN; this significantly slows down the LAN transfers.

12.11 SystemView OMEGAMON Services/400 Tips and Techniques

OMEGAMON monitoring and Automated Facilities/400 provide a set of powerful tools. However, if many conditions are defined and the situations containing these conditions are examined every few seconds, excessive CPU utilization can result. The major performance considerations include:

- Monitor only what is needed:

  Monitor conditions that are determined to be important and possible to occur during a period of time. For example, if a condition or situation can happen only during the nightly batch run, you may want to stop the monitoring at the beginning of the day until the nightly run time.

- Use embedded situations wherever possible:

  An attribute is a system condition that OMEGAMON/400 can monitor. For example, a job′s user profile is a condition, or a total number of jobs in the system is a condition. Conditions can be compared, summed, averaged, and so on. A situation is either a predicate (condition and compare, sum, average, and so on) or made up of other situations.

  An *embedded situation* is the situation that makes up another situation. By embedding situations where reasonable, OMEAGMON has less overhead by having to set up for the ″compound situation″ and not the individual situations.

- Minimize the starting and stopping of situations being monitored:

  Do not keep starting and stopping monitoring every few minutes.

- Minimize the monitoring intervals for specific conditions: A general guide for monitoring intervals is no sooner than every four minutes.
- Limit the amount of data sets gathered:

  ″Multiple instance″ attributes can cause a large number of sets of data to be gathered. For example, ″monitoring spooled files″ collects one set of data for each output queue on the system. The redbook *Managing Operations on AS/400s with SystemView OMEGAMON Services/400*, GG24-4136, lists high to low CPU impact multiple instance attributes. Performance impact increases if you do not specify compare values for one or more important attributes.

12.12 AS/400 RPG Tips and Techniques

ILE RPG/400 (RPG IV) offers many new ″built-in″ functions and ILE capabilities for using *call bound (static call)* programs (to improve performance) and grouping programs and associated files into activation groups. ILE tips and considerations are discussed in this section where they apply. If there are considerations unique to ILE, they are called out.

Note that most ILE considerations for activation groups, passing variables, and service programs discussed within Section 12.2, “General Application Programming Tips” on page 230 apply to RPG IV as well.

12.12.1 ILE RPG IV Built-In Functions

RPG IV contains many extensions to ″maximum values″ supported, such as character field length, constant data length, data structure size, and array size. Using these facilities should be considered in new program development to reduce cases where multiple definitions were required with RPG III.

RPG IV provides significant new support in time and date formats and compare and arithmetic operations with this new time and date support. There also is a new ″EVALuation″ opcode that supports familiar arithmetic operations and string concatenations in a single statement.

Use these new capabilities in new applications where tedious programming was required to achieve the same results, if possible.

12.12.2 ILE RPG IV Call Bound Support

The new CALLB operation supports the ILE bound call (static call) support for improved call performance compared to dynamic call support available under RPG III. This call bound support also includes defining data structures to be known outside of the program through the IMPORT and EXPORT keywords.

Before using the new CALLB support, you must understand the generally applicable ILE support constructs discussed in Section 12.2, “General Application Programming Tips” on page 230.

12.12.3 ILE RPG Program Use of Last Record (LR)

Using SETON LR in an RPG IV procedure running in a named activation group leaves the program activated but not invoked. While file Open Data Paths (ODPs) are closed, static storage is not freed and it is simply reinitialized the next time the procedure is called.

With an RPG III program, LR on also caused static storage to be freed up and returned to the system.

12.12.4 ILE RPG Program Size Consideration

In V3R6, there are several architectural factors that influence the size of program objects. First, the page size has increased from 512 byte bytes to 4K bytes (4KB). The larger page size is important in making storage management algorithms more efficient as the size of main storage continues to increase. The high end model 530 supports up to 4GB of main storage, and this may increase in future announcements. However, the 4KB page does impact the size of the objects, particularly smaller objects, since objects must be aligned on 4KB boundaries rather than 512 byte boundaries.

Second, in general, the number of instructions for a comparable program is going to be larger on RISC than on CISC. This is referred to as code expansion. By the nature of RISC design (efficient execution of simple instructions), it takes more instructions to do the same function as on CISC. For example, on RISC there are no storage-to-storage instructions; all data must be processed through registers. On CISC, moving data between two storage locations can be done with a single MVC (move character instruction). On RISC, this requires a Load and Store instruction. In most cases, code expansion has more of an impact on program object growth than the 4KB page size.

12.12.5 Run-Time Working Set Size

Due to the preceding factors, program object growth (for observable programs) when moving to V3R6 relative to V3R1 is as follows:

- ILE programs grow on the average by one to two times.
- Observable OPM programs grow on the average by two to three times.
- Non-observable OPM programs grow on an average by four to five times. This range may be as high as six times for small programs and as low as three times for large programs.

12.12.6 Compiler Working Set Size

In general, the ILE compilers and translators produce more internal tables of binding and debug information than their OPM counterparts. As a result, the compile time working set size has increased and compile times for large programs has also increased.

Working set size curves for a small-to-medium sized OPM RPG program are shown in Figure 49 on page 296. This program has 1500 C specifications and 5300 MI instructions. This program compiles reasonably fast using an 8MB pool if the program was not optimized. In V3R7, then optimizing part of the compiler was improved so that much less memory was required to achieve the same results as V3R6. In general, 12MB to 16MB as a minimum should be allocated to an ILE compiler especially if the *OPTIMIZE opion is selected.

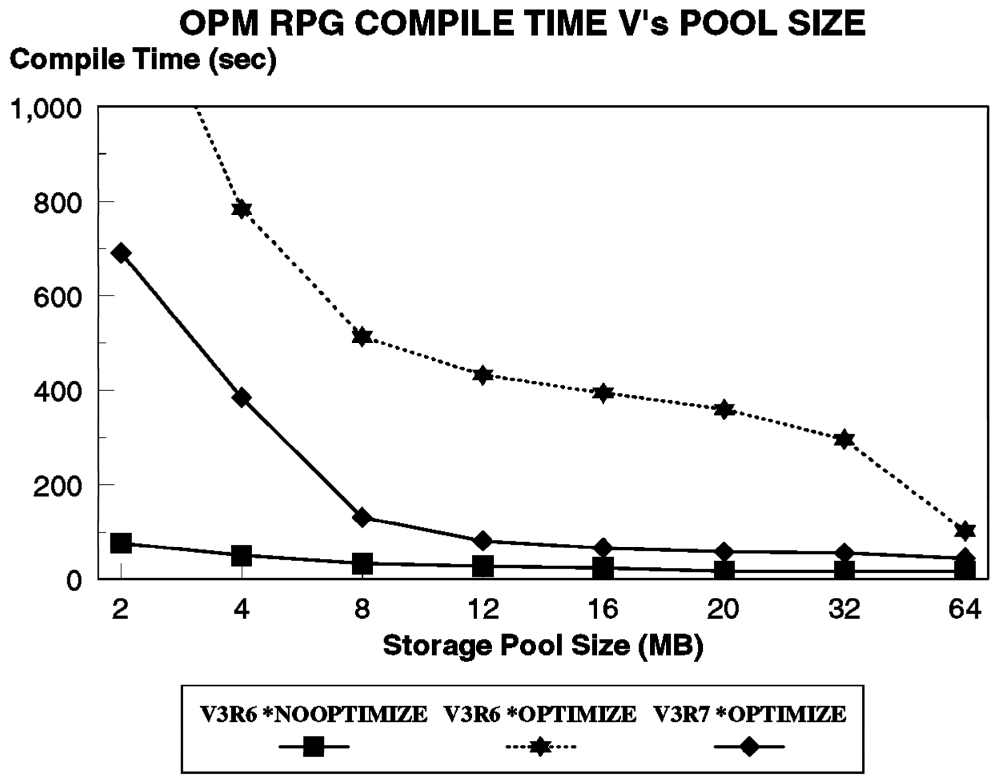


Figure 49. Working Set Size of OPM RPG Compiler

12.12.7 Compiler Options

Review the following compiler option before compiling your application programs.

Specify FIXDECDTA(*NO): On the Create System/36 RPG Program (CRTS36RPG) command. This prohibits the run-time routines from being called to correct invalid numeric data. This can be significant if the program performs many arithmetic operations.

*NO should be used after any decimal data problems with migrated System/36 files have been resolved.

OPTIMIZE on the Create RPG Program, Create Module Commands: For RPG III, specify *OPTIMIZE on the Create RPG Program command.

For ILE RPG IV, consider OPTIMIZE(*BASIC or *FULL) on the CRTxxxMOD command or CRTBNDRPG command. *BASIC costs less at compile time and results in small, but measurable performance improvement. In programs that are I/O intensive, *FULL shows little, if any performance improvement over *BASIC.

This invokes the machine interface translator to spend extra time analyzing the program to attempt to do additional optimization of pointer addressing for both local code generation and global pointer assignment.

For ILE, additional optimization of unused code is also performed.

Decimal Data Errors: The AS/400 system validates the data in numeric fields. If the field is in error, MCH1202 is signalled to the AS/400 RPG program. The CRTRPGPGM supports the ″ignore decimal data error″ IGNDECERR parameter. Specifying *YES enables the program to run, but degrades performance if many decimal errors are detected by the system.

The proper resolution is to correct the fields that have invalid data.

12.12.8 Which RPG Features to Use

When programming in RPG, it is possible to get the same program function using different RPG language features. Some are more efficient then others. This section gives tips on which features to use.

12.12.8.1 Conditioned ″DO″ Statements

The AS/400 system does not optimize indicator usage. Therefore, a program should use conditioned DO groups and the IF-THEN-ELSE statements, which generate much more efficient code than the use of indicators. For example:

```
 *   INDICATOR PROGRAM
C              READ FILE1       90 (EOF)
C  90   REPT   COMP  ′1′        91 (EQUAL)
C              SETON            LR
C  90 91       SETOF            LR
C  90 91′0000′ CHAINFILE1

 *   IMPROVED PROGRAM
C              READ FILE1       90 (EOF)
C  90          DO
C       *IN91  IFEQ  ′1′           (REPEAT ?)
C       *LOVAL CHAINFILE1
C              ELSE
C              SETON            LR
C              END
```

System/36 RPG supported indicator optimization. Therefore, migrated RPG programs achieve better AS/400 performance if changes to IF-THEN and DO statements are made.

12.12.8.2 Remove Invariant Variables and Subscript References

Removing variables from within a loop, especially if they are indexed, can save a significant amount of time.

Whenever a variable is referenced with a subscript, code must be generated to calculate the subscript.

```
 *   OLD PROGRAM
C              DO    100   IX    30
C       ARY,IX COMP  ′1′     888890 (EQUAL)
C  90          Z-ADD 1     I     30
C  90          ADD   1     Q,1
C  88   ARY,IX COMP  ′2′     888891 (EQUAL)
C  91          Z-ADD 2     I
C  91          ADD   2     Q,2
C              Z-ADD TOT   RTOT  30 (Statement is invariant)
```

```
C              END
 *   NEW PROGRAM
C              Z-ADDQ,1    TEMP1 30 (Subscript calc. removal)
C              Z-ADDQ,2    TEMP2 30
C  90          DO    100   IX    30
C       ARY,IX IFEQ  '1'
C              Z-ADD 1     I     30
C              ADD   1     TEMP1
C              ELSE
C       ARY,IX IFEQ  '2'
C              Z-ADD 2     I
C              ADD   2     TEMP2
C              END
C              END
C              END
C              Z-ADD TEMP1 Q,1
C              Z-ADD TEMP2 Q,2
C              Z-ADD TOT   RTOT  30 (Statement is invariant)
```

12.12.8.3 Return From a Subprogram

When a program and subprogram have a repetitive call/return sequence, it is usually more efficient for the subprogram to return to the called program with the LR (Last Record) indicator *off*. With LR off, the RETRN operation (op) code allows a subprogram to remain active (all files remain open, file position retained, and data values retained) across a return to the calling program. When this subprogram is called again, all file parameters and data values are as they were before the return. This minimizes system overhead but may require additional coding if some values need to be initialized each time the subprogram is called.

Note that use of this RETRN-with-LR-off technique should be used in a disciplined manner. If many programs remain active and their files open, this may cause an overall increase in system paging rates. The Reclaim Resource (RCLSRC) command may be used in the controlling program of a set of programs to free up (close) file resources below the program containing the RCLRSC command or the program that called this program with the RCLRSC command.

12.12.8.4 Sequential-Only Processing

If you know that the file is only going to be processed by this program or any called programs in a sequential manner, use SEQONLY(*YES) on an Override Database File (OVRDBF) command or Open Database File (OPNDBF) command before opening the file. In RPG, you must define the file for sequential input or sequential output only processing. Using either the CHAIN op code or the SETLL (Set Lower Limits) op code causes RPG to open the file randomly and SEQONLY(*YES) is ignored.

Note: You **must** close the file or issue the FEOD (Force End of Data) operation to get the last records written out to disk for an output file that is being processed as sequential only.

See Section 12.3, “Database Tips” on page 242 for more information on sequential-only processing.

12.12.8.5 QCLSCAN, QDCXLATE and RPG Data Scanning and Translate Functions

RPG has enhanced data scanning capabilities through the following operation codes: CAT, CHECK, SUBST, and XLATE. These functions should be used where possible rather than older MOVEA and LOKUP functions and should be considered as alternatives to the OS/400 functions provided by the QDCXLATE program and QCLSCAN program.

At this time, there are no performance comparisons between the OS/400 functions and the new RPG functions. The following example shows the call to QCLSCAN:

```
CALL ′QCLSCAN′
PARM                character string
PARM                string length
PARM                starting position
PARM                pattern character string
PARM                pattern length
PARM                xlate
PARM                trim
PARM                wild card
PARM                result field
```

Refer to the *OS/400 CL Programming V3R6*, SC41-4721, for more information.

If you have a large array that is ordered, consider using the Binary Sort program in optional library QUSRTOOL. QUSRTOOL can be migrated from V3R1, but certain functions may not perform as expected. It is not available in V3R6 except as a fee option. This program can significantly reduce CPU utilization.

12.12.8.6 OPEN Read-Only Files for Input-Only

If an application is working with a database file and has that file open for U-F (Update-full function), every file ″read″ (CHAIN, READ, READE, and so on) causes the database to lock the record just read. If the program does not require the record to be locked, it is wasted overhead and can actually slow other programs down by locking them out of the use of that record.

Therefore, open the file for input-only when no update or add record operations are performed.

12.12.8.7 For Files Opened for Update, Consider Record ″No Lock″ and ″Unlock″

RPG provides ″no lock″ and ″unlock″ functions. If a file is opened for U-F (Update-full function), a ″read″ operation (CHAIN, READ, READE, and so on) normally locks the record. If a specific read operation does not require the record to be locked, use an ″N″ in column 53 (″no lock″). This ″read″ places no lock on the record.

The RPG unlock (UNLCK) operation supports unlocking a database record as well as a data area. UNLCK for a database record should be considered when a workstation application may hold a lock for an extended period while waiting for operator input. A workstation application can timeout through the display file WAITRCD support if RPG Multiple Device File support is specified and a ″Write-with-DDS-Invite followed by Read-filename″ sequence is used. When the program detects the display file timeout, the program can issue the UNLCK operation for the database record.

Opening a file for input-only when no records are to be updated or using the Version 2 ″read with no lock″ and ″unlock″ operations on files opened for update should be considered in applications where long duration record locks have been identified as problem areas. For additional considerations on WAITRCD processing and all RPG methods of unlocking records, see the *RPG/400 User′s Guide*, SC09-1816.

12.12.8.8 Sorting of Tables - LOKUP

If a table is first sorted in ascending order, the high indicator can be specified in addition to the equal indicator to exit the search without going through the entire table.

Add records from the end towards the front and start LOKUP from the newest entry of the table.

12.12.9 Features to Avoid

In contrast to the previous section, here are some programming features to avoid.

12.12.9.1 DEBUG Operation Code, ILE DEBUG Option

For non-ILE programs, the AS/400 system provides a full symbolic debugger. It is not necessary to use the RPG language debug features. The use of the DEBUG opcode causes the compiler to generate additional code and data to accomplish the debugging function.

For ILE programs, ILE provides its own advanced debug support compared to the standard OS/400 symbolic debug support. Once the program has been fully debugged, we recommend removing debug from the ILE program by using the following command:

```
CHGPGM PGM(your program) RMVOBS(*DBGDTA).
```

12.12.9.2 Size Exceptions

The RPG language specifies that size exceptions are ignored. A size exception is when a value is too large to fit in the storage (such as adding one to a two-digit number that has the value 99). Handling the size exceptions does require additional processing even if they are ignored. Occurrence of size exceptions can be found in the Performance Tools Component Report under the heading *Exception Occurrence Summary and Interval Counts.*

12.12.10 System Features to Help Performance

12.12.10.1 CHGPGM OPTIMIZE(*YES or *FULL)

If you create a program without optimization, this command performs the optimization on the program. Use optimization on large programs that are called frequently. The optimize parameter has different alternatives depending on the compiler and language being used.

Optimization on large programs can take quite some time to complete.

12.12.10.2 Command Definitions to Invoke Programs

AS/400 command definition facility allows the programmer to define, in a controlled fashion, the parameter values passed to a program. You can specify the data types, values, ranges of values, and even default values for common parameters to the program. The use of a command definition can simplify program logic because a program can be written knowing the data is valid.

12.12.10.3 Avoid Using FREE in RPG Programs

Using FREE in RPG programs can result in disconnected file control blocks. Never use FREE with unshared (SHARE(*NO)) files.

12.12.10.4 Clearing Arrays

To clear an array in RPG, use a data structure and the CLEAR operation code.

12.12.10.5 Use Packed Decimal Odd Length Fields

The AS/400 system processes numeric fields most efficiently if they are defined as odd length packed decimal fields. This consideration is important only when the program performs frequent arithmetic operations.

Other systems may process other numeric field types more efficiently. For example, System/36 uses zoned decimal arithmetic. Since the System/36 RPG compiler on the AS/400 system (CRTS36RPG) **does not** generate packed decimal fields, you must use the CRTRPGPGM to get packed decimal field code generated for running on an AS/400 system.

12.12.10.6 Using Data Structures for Moves

Have file input or output fields in a data structure and do a single move to an array. A single move of a data structure is faster than moves for each field.

12.12.10.7 Use CHAIN Rather Than SETLL Operations

Whenever possible, use the CHAIN operation rather than SETLL. Set Lower Limits involves more system overhead. Note that using either CHAIN or SETLL causes RPG to open the file for random processing, which means any database file command SEQONLY parameter is ignored.

12.13 AS/400 COBOL Tips and Techniques

Although there are no specific ILE COBOL tips and techniques in this redbook, you should review the RPG section (specifically, Section 12.12, “AS/400 RPG Tips and Techniques” on page 294) for ILE compile time, compiled program size, compile time, and run-time working set size, program optimization, and program return (RPG SETON LR, COBOL STOP RUN) without closing files considerations.

The ILE considerations are almost identical for both RPG and COBOL.

Note that in general, COBOL and RPG Original Program Model (OPM) program performance is almost identical. COBOL is slightly faster in excessive disk I/O and batch applications. RPG is slightly faster in interactive applications. Comparison between ILE applications was not complete at the time of this redbook publication.

12.13.1 Compiler Options

GENOPT(*NORANGE): This option can be used to eliminate much of the range checking code that is generated to assure that the range of a subscript or index variable falls within the range of the dimension associated with the declared array. It also eliminates some of the checking code to assure that a varying-length character string move fits within bounds. This option **should not** be specified when you are testing/debugging the code and **should not** be used if you are unsure of the quality of your subscript variables; you **must know** that they remain within the range of the table definition.

OPTION(*NOXREF): Save the overhead of creating the symbol cross-reference listing.

GENOPT(*NOUNREF): This option informs the compiler **not** to place, in the debug symbol table, the names of program variables that are not referenced in the PROCEDURE DIVISION of the program. Using this feature does limit your ability to display program variables under debug for the items that are not referenced. This option can improve translate time somewhat because the translator does not have to process the extra object definition table entries. This can save some storage for programs that use many common copy books and do not reference many of the fields.

OPTION(*NOSOURCE): If you do not need the compiler listing every time you compile, you can save some time here. A general recommendation is that you always produce a listing so you can get to the statement numbers to do debugging.

12.13.2 Numeric Data Considerations

When programming in COBOL on an AS/400, there are a number of considerations to take into account when choosing the most appropriate numeric data representation.

12.13.2.1 Usage is Computational

The ANSI standard states that this data type is mapped to the most efficient commercial data support for the specific machine. The following list shows what COMPUTATIONAL maps do:

- S/3, S/32, S/36 - Zoned Decimal, same as USAGE IS DISPLAY.
- S/38, AS/400 - Packed decimal, same as USAGE IS PACKED-DECIMAL (COMP-3).
- S/370 - Binary, same as USAGE IS BINARY (COMP-4).

Because packed decimal is used on the AS/400 system, there are some things that should be considered when declaring COBOL variables. The machine stores packed decimal data in byte-aligned fields and most efficiently processes odd length packed decimal fields. The recommendation here is that ALL COMP-3 fields are declared where possible to be an odd number of decimal digits.

If you are migrating COBOL programs from a System/36, it is strongly recommended that you change heavily used computational items to be packed decimal. A very simple way to do this is to declare the numeric items as USAGE IS COMPUTATIONAL and the compiler chooses the appropriate data to use. Remember the CRTS36CBL command *does not select packed fields. You must* *use the CRTCBLPGM command for AS/400 (or the System/38) COBOL to achieve the best performance*. The overhead you pay for using zoned decimal on the AS/400 system is the additional conversions to and from packed decimal so the arithmetic operation can be performed (which equates to approximately three times as many instructions).

12.13.2.2 Signed versus Unsigned Arithmetic

All numeric PICTURE clauses can only contain the symbols 9, P, S, and V. The number of digits must range from one to 18 inclusive. If unsigned, the contents of the item in standard data format must contain a combination of Arabic numerals zero through nine. If signed, it may also contain a plus (+), minus (-), or other representation of the operational sign.

```
   01  A  PICTURE 999  USAGE IS COMP-3 VALUE 12.
   01  B  PICTURE S999 USAGE IS COMP-3 VALUE 12.
```

Whenever the compiler does a store into the variable A, extra code has to be generated to force the sign to a positive value even though the result is always positive. Unless you depend on the compiler changing a negative number into a positive value (absolute value concept), you should define **all of your data items as signed**. You can save the generation of one extra hardware instruction on every assignment.

12.13.2.3 Local Copies of Parameters

If you have parameters passed to a program, and heavily use the parameter within the program, move the parameter to a local workfield in *WORKING STORAGE*. If the parameter is numeric, have the Working Storage field defined as packed (COMP-3) of odd length.

If the data must be returned to the calling program, copy the Working Storage field to the parameter before returning control.

12.13.3 AS/400 versus S/36 Environment COBOL Differences

One of the most common problems for people migrating to the AS/400 system from the S/36 is that arithmetic on the AS/400 system is performed in packed decimal (similar to the S/370). Data such as blanks and other non-numeric data when encountered in data items causes an MCH1202 exception when used in numeric assignment, arithmetic, or compares. The S/36E COBOL compiler (CRTS36CBL) has an option, FIXDECDTA, which causes the compiler to generate extra code to attempt to fix **some** of these problems. The AS/400 COBOL compiler has no such option and program code and DDS may have to be changed to make the programs work the same as they did on the S/36. The INITIALIZE verb can be used to cause proper values to be placed in a group structure.

Note that for best performance, the data in error should be corrected.

12.13.4 Features to Use

When programming using the COBOL language on an AS/400, there are some features which you should consider to make efficient use of AS/400 resources.

12.13.4.1 Reference Modification

You should always consider using reference modification to work with substrings of character data. If you have the requirement to work with substrings (portions of a PIC X(nnn) character string), you should consider using the reference modification support. This is an ANSI 85 feature.

```
01  I  PIC S9(4) BINARY.
01  J  PIC S9(4) BINARY.
01  L  PIC S9(4) BINARY.
 . . .
MOVE S1(I:L) TO S2(J:L).
```

Reference modification, for example, enables you to move a 10-character string nine times faster than moving the 10 characters one byte at a time. Greater savings can be obtained with larger strings.

12.13.4.2 Inline PERFORM

Using inline PERFORM for looping is more efficient (approximately three times faster for the loop control code) than using out-of-line PERFORM. The compiler does not have to create linkage code to and from the performed paragraph.

```
   PERFORM VARYING A FROM 1 BY 1 UNTIL (I > 10)
       . . . code for paragraph P1
   END-PERFORM.
```

is much better than

```
   PERFORM P1 VARYING A FROM 1 BY 1 UNTIL (I > 10).
       . . .
  P1.
       . . . code for paragraph P1
```

12.13.4.3 GO TO DEPENDING ON versus IF-THEN

```
   GO TO P1, P2, P3, P4 DEPENDING ON I.

   versus

   IF I = 1 THEN GO TO P1
   ELSE IF I = 2 THEN GO TO P2
   ELSE IF I = 3 THEN GO TO P3
   ELSE IF I = 4 THEN GO TO P4.
```

If the value used to control the flow of control in the program represents a number whose range starts at one, it is usually more efficient to use the GO TO DEPENDING ON feature of the language. One of the more common uses for this might be in the testing of command keys pressed by the user. You can use the GO TO (even if there are gaps in the number values) by choosing an error procedure to branch to if the value is not the one desired.

```
   GO TO P1, ERROR-X, P3, P4 DEPENDING ON I.
```

The preceding example can be used if the valid values for I are 1, 3, or 4; and 2 is considered in error.

12.13.4.4 EXIT PROGRAM without CANCEL

EXIT PROGRAM allows a subprogram to remain active (all files can remain open, file position retained, and data values retained) across calls to the program. If the subprogram is dependent on the values in working storage being re-initialized at every call, you may want to consider adding additional program code to do this rather than issuing a STOP RUN which causes ALL programs in the run unit to be terminated. The extra code is not really much different from the code the compiler has to generate to do the initialization and you can do the initialization on a selective basis.

Note that COBOL standards dictate that the highest invocation (first COBOL program called) COBOL program is a run unit. When that program returns control, the files are closed and current data values are lost. On the AS/400 system, this can result in this EXIT technique being treated as a CANCEL, which means file open and variable initiation overhead occurs when the program is called again. On the AS/400 system, this can happen when a program written in CL or RPG receives control from the EXIT PROGRAM.

See Section 12.13.4.8, “COBOL MAIN Program Prior to Primary Program/Menu” on page 306 for a programming technique to address this possibility.

12.13.4.5 Sequential Only Processing

COBOL does automatic blocking/deblocking for a file if all of the following statements are true:

- ACCESS IS SEQUENTIAL
- File OPENed INPUT or OUTPUT
- ASSIGN to DISK, DATABASE, DISKETTE, TAPE
- No START statement is used for the file.

If you know that the file is only going to be processed by the called programs in a sequential manner, you should use SEQONLY(*YES) on the Override Database File (OVRDBF) command or Open Database File (OPNDBF) command preceding the COBOL OPEN file. This ensures record blocking is used to maximize throughput. See Section 12.3, “Database Tips” on page 242 for more information on sequential-only processing.

Note that a file opened for OUTPUT must be closed to have the last buffer of records written to disk.

12.13.4.6 Sequential-Only Processing Beginning with Version 2 Release 2

Additional sequential blocking support is provided if *all of the following conditions are true*:

- Either GENOPT(*BLK) is specified on the CRTCBLPGM command or the source program PROCESS statement specifies BLK.
- The source program File Description (FD) entry specifies BLOCK CONTAINS blocking values.

  The value 0 lets the system decide the blocking factor.
- ACCESS IS SEQUENTIAL OR ACCESS IS DYNAMIC is specified for the file.
- The file is opened for INPUT or OUTPUT in that program.

- The file is assigned to DISK, DATABASE, DISKETTE, or TAPEFILE.

When the preceding conditions are all met, COBOL performs blocking for the following file organizations and processing methods:

- A START statement followed by either READ PRIOR or READ NEXT
- INDEXED file having DYNAMIC ACCESS
- RELATIVE file having DYNAMIC ACCESS and opened for INPUT

Refer to the *COBOL/400 User′s Guide*, SC09-1812, for more information.

12.13.4.7 READ versus READ INTO

The AS/400 native COBOL record area associated with the FD definition is the actual file buffer. If you do not need to make a copy of the record in working storage, DO NOT use a ″READ file INTO record″ COBOL statement. This is a good recommendation for all files, especially sequential files.

12.13.4.8 COBOL MAIN Program Prior to Primary Program/Menu

The ANSI standard states that the highest level COBOL program in the run unit is considered the main program and any attempt to exit that program through an EXIT PROGRAM is to be considered a STOP RUN. A STOP RUN in COBOL causes the main program and all of the programs called by that program in the entire process to be canceled (CANCEL statement), which causes all of the files to be closed. To avoid this and still use a CL or RPG primary/driver program or menu to call a set of COBOL programs that you want to terminate using EXIT PROGRAM, create a simple COBOL program that just has a CALL ″cl-program/rpg-program″ statement in the PROCEDURE DIVISION.

This small program is now considered the main program of the run unit and all COBOL programs called by the driver program/menu are now considered subprograms. These subprograms can now all terminate using EXIT PROGRAM and the files can remain open across menu transitions (see Section 12.13.4.4, “EXIT PROGRAM without CANCEL” on page 305.)

12.13.4.9 QCLSCAN

Consider using QCLSCAN to replace some of the STRING and INSPECT functions normally coded in COBOL. This program has more functions and can actually perform faster than the COBOL support.

```
   CALL ″QCLSCAN″ USING CHAR-STRING, STRING-LENGTH,
                        START-POS, PATTERN, PATTERN-LENGTH,
                        XLATE, TRIM, WILD-CARD, RESULT-FIELD.
```

12.13.4.10 OPEN INPUT for Read-Only Files

If an application is working with a database file and has that file open for I-O (input-output), every file READ causes the database system to lock the record just read. If the program really does not need the record locked, it is a good AS/400 practice to have two different FDs in the program where one is open for INPUT and the other is open for I-O.

12.13.4.11 READ with ″NOLOCK″ for Input-Output Files

There is an alternative to using two files (FDs). If the file is open for I-O and you read a record, it is locked so that other users (JOBS) cannot get access to the record. If you know that you do not need the lock, you may want to do the following to free up the lock:

- For example:

  READ . . . WITH NOLOCK

  The ″WITH NOLOCK″ clause does not lock the record but since this file is open for I-O, the read waits if another job has the record locked.

12.13.4.12 Working Storage Values

A common programming problem dealing with workstations is that the programmer does not initialize working storage areas being displayed on the terminal. Characters whose value is less than a blank are, for the most part, control characters and can cause errors to occur for the workstation. The guideline here is that you either assign to all fields valid data or have a value assigned using the VALUE clause. You can also MOVE SPACES TO a record area to fill the entire record with spaces. Another approach is to use the INITIALIZE statement to set all the fields.

Note: SPACES are not valid zoned decimal numbers and cause an exception when used unless they are set to a valid number.

12.13.4.13 DECLARATIVES

When working with a display file (TRANSACTION file), it is a good practice to code a declarative section to handle error situations not handled by the program. Many people code a READ statement and do not code an AT END clause. You may get a file status ″10″ (end of file) set when the Roll key is pressed repeatedly; if there is no AT END clause on the READ, control is transferred to the DECLARATIVES. After execution of the declarative code, you can test the FILE STATUS or even the secondary FILE STATUS (major minor codes) to see what the exact error was. See Section 12.13.8, “Requesting a Formatted Symbolic Dump” on page 316 for an example of coding a declarative section.

12.13.4.14 Sorting Tables

Often a table is built and searched to find an entry in the table. If the table is going to be searched often, it is a good idea to sort the table before you do the search. This allows using a SEARCH ALL (binary search) statement in COBOL. There are two sort algorithms coded as sample programs at the end of this section. The HEAPSORT is usually faster than the SHELL sort. See the coding examples in this chapter.

12.13.5 Features to Avoid

Here are a number of features which should be avoided when writing COBOL programs that will run on an AS/400.

12.13.5.1 STRING and UNSTRING

These functions are implemented as external run-time subroutines and, in many cases, simple MOVE statements where extra REDEFINES do the same function much more efficiently.

One STRING statement can be written instead of a series of MOVE statements.

The UNSTRING statement causes contiguous data in a sending field to be separated and placed into multiple receiving fields.

```
STRING  ID-1  ID-2  DELIMITED  BY ID-3
        ID-4  ID-5  DELIMITED  BY SIZE
  INTO  ID-7  WITH POINTER  ID-8
END-STRING
```

For more information on how this works, see Figure 22 on page 419 in *ILE COBOL/400 Reference*, SC09-2073.

Reference modification makes it nine times faster to move a 10-character string than moving the 10 characters one byte at a time. Greater savings can be obtained with larger strings.

12.13.5.2 ACCEPT and DISPLAY for Input-Output

Accept and display functions make use of the message handler services to send and receive messages to message queues. It is normally better performance to use files for data input or output operations. Extended Accept Display COBOL support (PROCESS statement EXTACCDSP parameter) gives improved performance and more control over positioning both input and output fields.

12.13.5.3 START on a Multi-Format File

Care should be taken when attempting to use a START statement with a partial key to position a file on a logical file that has multiple formats associated with it and the format specified is for records that do not occur too frequently in the file and there are a lot of other keys in the other formats. Performance can be quite slow to satisfy this request.

12.13.5.4 Segmentation

Using segmentation on the AS/400 system adds overhead to the program. This feature is provided for compatibility with other COBOL compilers and is not necessary given the AS/400 virtual storage model.

12.13.5.5 Overuse of Subprograms

CALL has significantly more overhead than a PERFORM. A general guideline to follow is to do a ″meaningful″ amount of work in the subprogram. Many times, you can create a COPY book that represents a standard function that can be included in each program that needs the function. Do not get carried away with creating ″large″ (greater than 5000-line) programs. Strive for functional packaging of units of code. Large programs take more time to compile and are harder for the compiler to optimize.

12.13.5.6 USE FOR DEBUGGING

Since the AS/400 system provides a full symbolic debugger, it is not necessary to use the COBOL language debug features. Using the USE FOR DEBUGGING feature causes the compiler to generate a lot of additional code and data to accomplish the function.

12.13.5.7 Read Under Format

It is recommended that if you use the design approach where you have one program simply open the workstation, write and invite input, and close the file, you have the next program open the workstation and read the format written by the last program. You may want to combine these two programs into one program. This technique was often used by S/36 application programs.

12.13.5.8 CANCEL

Using the CANCEL verb causes all program data and all files associated with the program being canceled to be deactivated. This means the files are closed and copying the static storage in the process is destroyed and removed from the process. Subsequent calls to the canceled program cause the system to re-establish the program environment within the process, which is significant overhead. See ″EXIT PROGRAM without CANCEL″ for alternate ways to accomplish the function.

12.13.5.9 Exponentiation - ″**″ Operator for S/36 or S/38 COBOL

The AS/400 COBOL compiler generates inline code for the exponentiation operator. The S/36 and S/38 COBOL compilers currently generate this code using an external procedure. One common use of this is to take the square root of a number. This can be done efficiently inline using Newton′s approximation method for square root. It is more efficient to square or cube a number by using consecutive MULTIPLY operations rather than using the exponentiation operator. See the sample program for inline square root using COBOL.

12.13.5.10 Size Exceptions

The COBOL language specifies that size exceptions are ignored unless the programmer has an ON SIZE ERROR clause associated with the arithmetic operation. Handling the size exceptions does require additional processing even if they are ignored. You can determine if the program had any size errors under debug by DSPPGMVAR ′.MGTCNTR(1)′.

Note: The Performance Tools Component Report indicates the frequency of exceptions on a system-wide basis.

12.13.6 Handy Things to Know

Here are some handy things to know when programming in COBOL on an AS/400.

12.13.6.1 COBOL File Information Block

Even though you receive an error message at compile time, the AS/400 COBOL compiler allows you to pass a file variable as a parameter to another program. What is actually passed by the compiler when you pass the FD-name is an internal compiler control block called the File Information Block (FIB). This control block contains a lot of information about the file and the state of the file that can be useful in debugging some I/O related problems. You can write a COBOL subprogram to print or display the FIB.

```
*---------------------------------------------------------------*
*  COBOL FILE CONTROL BLOCK - GENERATED BY COMPILER FOR FD      *
*   Note: This control block is subject to change by the        *
*         compiler at release boundaries and may require        *
*         you to recompile if you use it.                       *
*---------------------------------------------------------------*
 01  FIB.
     05  FIBFN PIC X(30).
     05  FIBALT PIC X.
*        BIT 1 -> CLEAR FILE
*        BIT 2 -> INITIALIZE FILE TO DELETED RECORDS
*        BIT 3 -> OPTIONAL FILE NOT FOUND
*        BIT 4 -> FOOTING VALUE FOR LINAGE DEFAULTED
*        BIT 5 -> BYPASS DUPLICATE CHECKING
*        BIT 6 -> NON DELETE CAPABLE FILE
*        BIT 7 -> PAGE OVERFLOW
*        BIT 8 -> OPEN REVERSED IN EFFECT
     05  FIBFLGS PIC X.
*        BIT 1 -> FILE IS OPEN
*        BIT 2 -> FILE IS LOCKED VIA CLOSE LOCK
*        BIT 3 -> EOF HAS OCCURRED
*        BIT 4 -> RESERVED
*        BIT 5 -> OPTIONAL FILE
*        BIT 6 -> CHECK INDEXED FILE FOR DUPLICATES AT OPEN
*        BIT 7 -> END OF PAGE HAS OCCURRED
*        BIT 8 -> START REL LAST OP
     05  FIBCUR.
         10  FIBCOP PIC X(4).
*        CURRENT FILE STATUS
         10  FIBCFS PIC X(2).
*        OLD FILE STATUS
     05  FIBOFS PIC X(2).
*        COBOL VERB USED ON LAST I-O
     05  FIBVERB PIC S9(4) BINARY.
         88  CLOSE-VERB                VALUE 1 .
         88  DELETE-VERB               VALUE 2 .
         88  OPEN-VERB                 VALUE 3 .
         88  READ-VERB                 VALUE 4 .
         88  REWRITE-VERB              VALUE 5 .
         88  START-VERB                VALUE 6 .
         88  WRITE-VERB                VALUE 7 .
         88  COMMIT-VERB               VALUE 8 .
         88  ROLLBACK-VERB             VALUE 9 .
         88  ACQUIRE-VERB              VALUE 10.
         88  DROP-VERB                 VALUE 11.
     05  FIBSPC PIC X(14).
*    LINAGE INFORMATION FOR PRINT FD
     05  FIBLINAGE REDEFINES FIBSPC.
         10  FIBLIN   PIC S9(4) BINARY.
         10  FIBLFT   PIC S9(4) BINARY.
         10  FIBLTO   PIC S9(4) BINARY.
         10  FIBLBO   PIC S9(4) BINARY.
         10  FIBLINE  PIC S9(4) BINARY.
     05  FIBRELKEY REDEFINES FIBSPC.
         10  FIBKEY   PIC S9(9) BINARY.
         10  FIBKLEN  PIC S9(4) BINARY.
     05  FIBUSE       PIC S9(4) BINARY.
     05  FIBFMT       PIC X(10).
     05  FIBCRP       PIC X.
     05  FIBDEVC      PIC S9(4) BINARY.
         88 OTHER-DEVICE      VALUE 0.
         88 DISPLAY-DEVICE    VALUE 1.
         88 PRINTER-DEVICE    VALUE 2.
         88 CARD-DEVICE       VALUE 3.
         88 DISKETTE-DEVICE   VALUE 4.
         88 TAPE-DEVICE       VALUE 5.
     05  FIBORG PIC S9(4) BINARY.
         88 SEQUENTIAL-ORG   VALUE 1.
         88 RELATIVE-ORG     VALUE 2.
         88 INDEXED-ORG      VALUE 3.
         88 TRANSACTION-ORG  VALUE 4.
     05  FIBACC PIC S9(4) BINARY.
         88 SEQUENTIAL-ACC   VALUE 1.
         88 RANDOM-ACC       VALUE 2.
         88 DYNAMIC-ACC      VALUE 3.
     05  FIBOTP PIC S9(4) BINARY.
         88 OPENED-INPUT         VALUE 1.
         88 OPENED-OUTPUT        VALUE 2.
         88 OPENED-EXTEND        VALUE 3.
         88 OPENED-I-O-U-D       VALUE 4.
         88 OPENED-I-O           VALUE 5.
*        SHORTEST RECORD LENGTH USED FOR FILE
     05  FIBRECS PIC S9(4) BINARY.
     05  FILLER PIC X(01).
*        POINTER TO UFCB
     05  FIBUFCB PIC X(16).
     05  FILLER PIC X(32).
*        POINTER TO NEXT UFCB IN COBOL CHAIN
     05  FIBCHAN PIC X(16).
*        POINTER TO BUFFER POINTER
     05  FIBP1   PIC X(16).
     05  FIBDEVN PIC X(10).
     05  FIBDEVI PIC S9(4) BINARY.
*        EXETENDED FILE STATUS
     05  FIBCFS2 PIC X(4).
     05  FILLER PIC X.
     05  FIBOLDK.
         10  FIBOKLN PIC S9(4) BINARY.
         10  FIBOKEY PIC X(121).
     05  FIBRELS REDEFINES FIBOLDK.
         10  FIBORRN PIC S9(9) BINARY.
         10  FIBCA PIC X(22).
*        TERMINAL INFORMATION
     05  FIBTERMS REDEFINES FIBOLDK.
         10 FILLER PIC X(4).
*           FUNCTION KEY ID
         10 FIBCKID PIC S99.
*           TERMINAL ID
         10 FIBCTID PIC X(10).
*           FORMAT NAME
         10 FIBCFMT PIC X(10).
*        OLD FORMAT NAME
     05  FIBOFMT PIC X(10).
*        MEMBER NAME
     05  FIBMBRN PIC X(10).
```

12.13.6.2 Cursor Position

You can obtain the cursor position of the field that was positioned to when the READ was completed through the I-O-FEEDBACK information for the file. Use the ACCEPT statement to retrieve this data. See the I-O-FEEDBACK sample data declares for COBOL layout.

```
   . . .
   SPECIAL-NAMES. I-O-FEEDBACK IS FILE-IO-FEEDBACK.
   . . .
   DATA DIVISION.
   . . .
   WORKING-STORAGE SECTION.
   01  LINE-POS PIC S999 PACKED-DECIMAL.
   01  COL-POS  PIC S999 PACKED-DECIMAL.
  * two byte binary field filled with all zeros
   01  WORK-BIN PIC S9(4) BINARY VALUE 0.
   01  WORK-X   REDEFINES WORK-BIN.
       02  FILLER PIC X.
  *        right-most byte of two byte binary field
       02  WORK-CHR PIC X.

  * INCLUDE THE I-O FEEDBACK COPY BOOK
   COPY IOFB.
   . . .
   PROCEDURE DIVISION.
   . . .
       ACCEPT IOF-DSP-AND-ICF FROM FILE-IO-FEEDBACK
          FOR DSP-FILE.
       MOVE IOF-DSP-CURSOR-LINE TO WORK-CHR.
       MOVE WORK-BIN TO LINE-POS.
       MOVE IOF-DSP-CURSOR-COL TO WORK-CHR.
       MOVE WORK-BIN TO COL-POS.
```

12.13.6.3 Did Data Entry Occur?

You can get COBOL to do an invite input on a write operation by using the INVITE keyword in DDS and defining your display file as MAXDEV(2), any number greater than one. This allows you to continue processing in your program while the user types in data. When the user presses Enter or a PF key that terminates data entry, the attribute data for the workstation file is updated.

You can also determine if data entry has occurred by checking a bit in the ATTRIBUTE-DATA. It is bit 5 at offset 76. This bit is set when you have an ″accept input″ pending for the device and the user presses Enter or some function key. You can test this bit without having to do the READ for the file. To test bit 5, you can use the following programming technique:

```
   PROCESS NOTRUNC.
   IDENTIFICATION DIVISON.
   . . .
   SPECIAL-NAMES. ATTRIBUTE-DATA IS FILE-ATTR-DATA.
   . . .
   DATA DIVISION.
   . . .
   WORKING-STORAGE SECTION.
   01  ANS      PIC S9(4) BINARY VALUE 0.
   01  REM      PIC S9(4) BINARY VALUE 0.
  * two byte binary field filled with all zeros
   01  WORK-BIN PIC S9(4) BINARY VALUE 0.
   01  WORK-X   REDEFINES WORK-BIN.
```

```
      02  FILLER PIC X.
 *        right-most byte of two byte binary field
      02  WORK-CHR PIC X.

 * INCLUDE THE ATTRIBUTE DATA COPY BOOK
  COPY IOATTR.
  . . .
  PROCEDURE DIVISION.
  . . .
      ACCEPT ATR-INFORMATION FROM FILE-ATTR-DATA
         FOR DSP-FILE.
  . . .
      MOVE ATR-BIT-FLAGS TO WRK-CHR.
*----------------------------------------------------------*
*     get the remainder after dividing by 16 (value 0..15) *
*     which is the value of bits 5,6,7,8 of the byte       *
*----------------------------------------------------------*
      DIVIDE WORK-BIN BY 16 GIVING ANS REMAINDER REM.

*----------------------------------------------------------*
*     any value greater than 7 says bit 5 is on            *
*----------------------------------------------------------*
      IF REM > 7 THEN
         you know that data available bit is now on
         for any code executed here
      END-IF.
```

12.13.6.4 Common Input-Output Handling

With the use of OPEN-FEEDBACK information and the ability to do dynamic file overrides with the AS/400 system, you can write a single COBOL program to manipulate any sequential file, relative file, or indexed file processed sequentially.

```
 IDENTIFICATION DIVISION.
   PROGRAM-ID. SEQTEST.
*  THIS IS AN EXAMPLE PROGRAM TO DEMONSTRATE THE ABILITY TO
*  WRITE A SINGLE PROGRAM TO HANDLE SEQUENTIAL FILES UP TO
*  A MAX RECORD SIZE (SET BY THIS PROGRAM) IN A S/36 ENV
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
   SOURCE-COMPUTER. IBM-AS400.
   OBJECT-COMPUTER. IBM-AS400.
 SPECIAL-NAMES.  OPEN-FEEDBACK IS FILE-OPEN-FEEDBACK.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT SEQ-FILE ASSIGN TO DATABASE-SEQFILE
            ORGANIZATION IS SEQUENTIAL
            ACCESS IS SEQUENTIAL
            FILE STATUS IS SEQ-FILE-STATUS.
 DATA DIVISION.
 FILE SECTION.
 FD  SEQ-FILE.
 01  SEQ-RECORD.
     02 SEQ-DATA PIC X
        OCCURS 1 TO 4000 TIMES DEPENDING ON RECL.
 WORKING-STORAGE SECTION.
 01  SEQ-FILE-STATUS    PIC XX VALUE ”??”.
 01  RECL               PIC S9(5) PACKED-DECIMAL.
 01  FORMAT-NAME        PIC X(10).
 01  CMD-LEN            PIC S9(10)V9(5) PACKED-DECIMAL VALUE 52.
 01  CMD.
     02 FILLER          PIC X(29) VALUE
        ″ OVRDBF FILE(SEQFILE) TOFILE(″.
     02 CMD-FILE        PIC X(10).
     02 FILLER          PIC X(13) VALUE
        ″) MBR(*LAST) ″.
* INCLUDE THE OPEN FEEDBACK COPY BOOK
 COPY OPENFB.
 LINKAGE SECTION.
 01  CONTROL-INFO.
     03  FILE-NAME      PIC X(10).
     03  FILE-RECL      PIC S9(5) PACKED-DECIMAL.
     03  ACTION-CODE    PIC X(5).
     03  RC             PIC XX.
 01  USER-BUFFER.
     02 USER-DATA PIC X
        OCCURS 1 TO 4000 TIMES DEPENDING ON RECL.

 PROCEDURE DIVISION USING CONTROL-INFO, USER-BUFFER.
 START-OF-PROGRAM.
*-----------------------------------------------------------------*
* TEST OPTIONS IN ORDER OF FREQUENCY OF OCCURRENCE FOR PERFORMANCE *
*-----------------------------------------------------------------*
     IF ACTION-CODE  = ″READ ″ THEN
        READ SEQ-FILE RECORD
           AT END
             GO TO END-OF-PGM
           NOT AT END
             MOVE SEQ-RECORD TO USER-BUFFER
        END-READ
        GO TO END-OF-PGM
     END-IF.

*-----------------------------------------------------------------*
     IF ACTION-CODE = ″OPENF″ THEN
        MOVE  FILE-NAME  TO CMD-FILE
*       OVERRIDE TO ACTUAL FILE THAT IS ON ″FILE″ OCL
        CALL ″QCMDEXC″ USING CMD, CMD-LEN
        OPEN I-O SEQ-FILE
        ACCEPT OFB-INFORMATION FROM FILE-OPEN-FEEDBACK
            FOR SEQ-FILE
*       EXTRACT RECORD LENGTH SO MOVES ONLY MOVE CORRECT AMOUNT
        MOVE OFB-REC-LENGTH TO RECL, FILE-RECL
        GO TO END-OF-PGM
     END-IF.

*-----------------------------------------------------------------*
     IF ACTION-CODE = ″CLOSE″ THEN
        CLOSE SEQ-FILE
        GO TO END-OF-PGM
     END-IF.

*-----------------------------------------------------------------*
* FALL THRU HERE INDICATES A BAD ACTION CODE WAS PASSED          *
*-----------------------------------------------------------------*
 ERROR-BAD-OP.
     MOVE ″90″ TO RC.
     EXIT PROGRAM.
```

```
END-OF-PGM.
    MOVE SEQ-FILE-STATUS TO RC.
    EXIT PROGRAM.
```

12.13.7 Sample COBOL Code to Do Square Root Inline

Here is a routine that calculates the square root of a number as inline code, using Newton's approximation method.

12.13.7.1 DDS for the Program

```
A                                      DSPSIZ(24 80 *DS3)
A          R SQRTFMT
A                                  1 20'COBOL Square Root Demonstration'
A                                      DSPATR(HI)
A                                  4 11'Number to take Square Root of:'
A                                  6 26'Square Root is:'
A            NBR            5Y 0B  4 43CHECK(RZ)
A                                      CHECK(ME)
A                                  9 11'Enter a zero (0) to terminate proc-
A                                      essing'
A            ANS            7Y 30  6 43EDTWRD('    .   ')
```

12.13.7.2 COBOL Program for SQRT

```
 IDENTIFICATION DIVISION.
   PROGRAM-ID. SQRT.
   AUTHOR. Richard L Bains.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
   SOURCE-COMPUTER.  IBM-AS400.
   OBJECT-COMPUTER.  IBM-AS400.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT WORK-STATION ASSIGN TO WORKSTATION-SQRTDS
            ORGANIZATION IS TRANSACTION.
 DATA DIVISION.
 FILE SECTION.
 FD  WORK-STATION.
 01  WS-RECORD.
 COPY DD-ALL-FORMATS OF SQRTDS.
 WORKING-STORAGE SECTION.
 01  SQRT-NBR  PIC S9(5)V99   PACKED-DECIMAL.
 01  SQRT      PIC S9(5)V9999 PACKED-DECIMAL.
 01  SQRT-C    REDEFINES SQRT   PIC X(4).
 01  SQRT-2    PIC S9(5)V9999 PACKED-DECIMAL.
 01  SQRT-2-C  REDEFINES SQRT-2 PIC X(4).
* NOTE: CHARACTER OVERLAY IS USED TO DETERMINE WHEN THE
*       ACCURACY IS REACHED FOR THE NUMBER.  IT OVERLAYS
*       ALL BUT THE LAST BYTE OF THE PACKED DECIMAL NUMBER.
 PROCEDURE DIVISION.
 START-OF-PROGRAM.
     OPEN I-O WORK-STATION.
     MOVE ZEROS TO WS-RECORD.
 PGM-LOOP.
     WRITE WS-RECORD FORMAT IS "SQRTFMT".
     READ WORK-STATION.
     IF NBR OF SQRTFMT-I = 0 THEN STOP RUN.
     MOVE NBR OF SQRTFMT-I TO SQRT-NBR.
     PERFORM SQUARE-ROOT THRU END-SQUARE-ROOT.
     MOVE SQRT-NBR TO NBR OF SQRTFMT-O.
     MOVE SQRT TO ANS OF SQRTFMT-O.
     GO TO PGM-LOOP.
**
* NEWTON’S APPROX METHOD: X(N+1) = .5 * (X(N) + NBR / X(N))
* LINKAGE IS : MOVE ?????? TO SQRT-NBR.
*              PERFORM SQUARE-ROOT THRU END-SQUARE-ROOT.
* OUTPUT: ANSWER IS IN ″SQRT″
**
 SQUARE-ROOT.
     MOVE SQRT-NBR TO SQRT.
     PERFORM 8 TIMES
       COMPUTE SQRT-2 = 0.5 * ( SQRT + SQRT-NBR / SQRT )
       IF SQRT-C = SQRT-2-C THEN
         GO TO END-SQUARE-ROOT
       END-IF
       MOVE SQRT-2 TO SQRT
     END-PERFORM.
 END-SQUARE-ROOT.
     EXIT.
```

12.13.8 Requesting a Formatted Symbolic Dump

You can force a symbolic dump to be triggered for a program by calling the COBOL run-time routine for the specific environment to get a listing of the files and the variables used in the program:

- S/38 -- program is QCREXHAN
- AS/400 -- program is QLREXHAN

12.13.8.1 Calling Dump from COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SQRT.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
  SOURCE-COMPUTER.  IBM-AS400.
  OBJECT-COMPUTER.  IBM-AS400.
SPECIAL-NAMES.
    UPSI-0 IS FORMATED-DUMP-SWITCH
        ON STATUS IS TAKE-THE-DUMP
        OFF STATUS IS DO-NOT-TAKE-THE-DUMP.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT SCREEN-FILE ASSIGN TO WORKSTATION-SCREENDS
           ORGANIZATION IS TRANSACTION
           FILE STATUS IS SCREEN-FS,
                          SCREEN-FS-MAJOR-MINOR.
DATA DIVISION.
FILE SECTION.
FD  WORK-STATION.
01  WS-RECORD.
COPY DD-ALL-FORMATS OF TERMFILE.
WORKING-STORAGE SECTION.
01  SCREEN-FS             PIC X(2).
01  SCREEN-FS-MAJOR-MINOR PIC X(4).
01  DUMP-PARMS.
    05  TYPE-OF-DUMP PIC X     VALUE ″F″.
    05  FILLER       PIC X     VALUE ″-″.
    05  PGM-NAME     PIC X(10) VALUE ″a name″.
```

```
 . . .
 PROCEDURE DIVISION.
 DECLARATIVES.
 SCREEN-ERROR SECTION.
     USE AFTER STANDARD EXCEPTION PROCEDURE ON SCREEN-FILE.
 SCREEN-ERROR-HANDLER.
* You can add anything you want here including a test for
* some global debug switch like the job UPSI switches
* that can be changed with a CHGJOB SWS(1XXXXXXX)
* In production mode you may want to write information from
* the terminal or a file or job log.
     IF TAKE-THE-DUMP THEN
         CALL ″QLREXHAN″ USING DUMP-PARMS
     ELSE
         DISPLAY ″Error occurred on SCREEN-FILE, File Status =″
                 SCREEN-FS ″/″ SCREEN-FS-MAJOR-MINOR.
 END DECLARATIVES.
 MAIN-PROCEDURE SECTION.
    START-OF-PROGRAM.
    . . .
```

12.13.9 Heap Sort Subroutine

```
 PROCESS NOTRUNC.
 IDENTIFICATION DIVISION.
   PROGRAM-ID. HEAPSORT.
   AUTHOR. Richard L Bains.
   INSTALLATION.  IBM Rochester.
*--------------------------------------------------------------*
* NOTES:                                                       *
*   The heap sort algorithm is usually slightly faster than    *
*   the Shell sort algorithm, but a slower than a quicksort.   *
*   this algorithm is a very stable and predictable performing *
*   algorithm.                                                 *
* Use GENOPT(*NORANGE) to get better performing code when      *
*     compiling this subroutine                                *
* Also note that NOTRUNC is used to obtain better subscripting *
* code.                                                        *
*--------------------------------------------------------------*
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
   SOURCE-COMPUTER. IBM-AS400.
   OBJECT-COMPUTER. IBM-AS400.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  N                                PIC S9(4) BINARY.
 01  LEFT-INDEX                       PIC S9(4) BINARY.
 01  RIGHT-INDEX                      PIC S9(4) BINARY.
 01  THE-FRONT                        PIC S9(4) BINARY.
 01  THE-FRONT-TIMES-2                PIC S9(4) BINARY.
 01  THE-BACK                         PIC S9(4) BINARY.
 01  SUCC-THE-BACK                    PIC S9(4) BINARY.
* TEMPORARY TO HOLD ONE ITEM FROM TABLE
 01  TEMP-ITEM.
     02 TEMP-KEY                      PIC X(5).
     02 FILLER                        PIC X(1).
 LINKAGE SECTION.
 01  NUMBER-OF-ITEMS                  PIC S9(4) BINARY.
 01  ITEM-TABLE.
```

```
      02  THE-ITEMS OCCURS 5000 TIMES.
          03  THE-KEY                  PIC X(5).
          03  FILLER                   PIC X(1).


  PROCEDURE DIVISION USING NUMBER-OF-ITEMS, ITEM-TABLE.
  START-OF-PROGRAM.
 * MAKE A LOCAL COPY OF NUMBER OF TABLE ENTRIES
      MOVE NUMBER-OF-ITEMS TO N.
      PERFORM HEAP-SORT.
      EXIT PROGRAM.
 *---------------------------------------------------------------*
 * HEAP SORT SUBROUTINE                                          *
 *---------------------------------------------------------------*
  HEAP-SORT.
*     INDEX = ( N / 2 ) + 1.
      DIVIDE N BY 2 GIVING LEFT-INDEX.
      ADD 1 TO LEFT-INDEX.
      MOVE N TO RIGHT-INDEX.
      PERFORM UNTIL LEFT-INDEX <= 1
        SUBTRACT 1 FROM LEFT-INDEX
        PERFORM SIFT THRU SIFT-EXIT
      END-PERFORM.
      PERFORM UNTIL RIGHT-INDEX <= 1
         MOVE THE-ITEMS ( 1 ) TO TEMP-ITEM
         MOVE THE-ITEMS ( RIGHT-INDEX ) TO THE-ITEMS ( 1 )
         MOVE TEMP-ITEM TO THE-ITEMS ( RIGHT-INDEX )
         SUBTRACT 1 FROM RIGHT-INDEX
         PERFORM SIFT THRU SIFT-EXIT
      END-PERFORM.
 *---------------------------------------------------------------*
 * SIFT ONE ITEM TO TOP OF THE HEAP                              *
 *---------------------------------------------------------------*
  SIFT.
      MOVE THE-ITEMS ( LEFT-INDEX ) TO TEMP-ITEM.
      MOVE LEFT-INDEX TO THE-FRONT.
 *    MULTIPLY BY TWO
      ADD THE-FRONT TO THE-FRONT GIVING THE-BACK.
      PERFORM UNTIL THE-BACK > RIGHT-INDEX
        IF THE-BACK < RIGHT-INDEX THEN
          ADD      1 FROM THE-BACK GIVING SUCC-THE-BACK
          IF THE-KEY ( THE-BACK ) <
             THE-KEY ( SUCC-THE-BACK ) THEN
             MOVE SUCC-THE-BACK TO THE-BACK
          END-IF
        END-IF
        IF TEMP-KEY >= THE-KEY ( THE-BACK ) THEN
          GO TO SIFT-MOVE
        END-IF
        MOVE THE-ITEMS ( THE-BACK ) TO THE-ITEMS ( THE-FRONT )
        MOVE THE-BACK TO THE-FRONT
        ADD THE-FRONT TO THE-FRONT GIVING THE-FRONT-TIMES-2
        IF THE-FRONT-TIMES-2 > N THEN
          GO TO SIFT-MOVE
        END-IF
        MOVE THE-FRONT-TIMES-2 TO THE-BACK
      END-PERFORM.
  SIFT-MOVE.
      MOVE TEMP-ITEM TO THE-ITEMS ( THE-FRONT ).
  SIFT-EXIT.
      EXIT.
```

12.13.10 Shell Sort Subroutine

```
 PROCESS NOTRUNC.
 IDENTIFICATION DIVISION.
   PROGRAM-ID. SHELLSORT.
   AUTHOR. Richard L Bains.
   INSTALLATION.  IBM Rochester.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
   SOURCE-COMPUTER. IBM-AS400.
   OBJECT-COMPUTER. IBM-AS400.
*--------------------------------------------------------------*
* NOTES:                                                       *
*   The Shell sort algorithm is another fast algorithm for     *
*   sorting in-storage tables.  It is somewhat smaller in code *
*   than the heap sort, but usually is slightly slower.        *
*   If the table is sorted in reverse order, you can get       *
*   worst case performance.                                    *
* use GENOPT(*NORANGE) to get better performing code when      *
*     compiling this subroutine                                *
*     note that NOTRUNC is used to obtain better subscripting  *
* code.                                                        *
*--------------------------------------------------------------*
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  I                                PIC S9(4) BINARY.
 01  J                                PIC S9(4) BINARY.
 01  K                                PIC S9(4) BINARY.
 01  INCR                             PIC S9(4) BINARY.
 01  T-INCR                           PIC S9(4) BINARY.
 01  N                                PIC S9(4) BINARY.
 01  TEMP-ITEM.
         03  TEMP-KEY                 PIC X(5).
         03  FILLER                   PIC X(1).
 LINKAGE SECTION.
 01  NUMBER-OF-ITEMS                  PIC S9(4) BINARY.
 01  ITEM-TABLE.
     02  THE-ITEMS OCCURS 5000 TIMES.
         03  THE-KEY                  PIC X(5).
         03  FILLER                   PIC X(1).


 PROCEDURE DIVISION USING NUMBER-OF-ITEMS, ITEM-TABLE.
 START-OF-PROGRAM.
     MOVE NUMBER-OF-ITEMS TO N.
     PERFORM SHELL-SORT.
     EXIT PROGRAM.
*--------------------------------------------------------------*
* SHELL SORT SUBROUTINE                                        *
*--------------------------------------------------------------*
 SHELL-SORT.
     MOVE 1 TO INCR.
     MULTIPLY INCR BY 9 GIVING T-INCR.
     ADD 4 TO T-INCR.
     PERFORM UNTIL T-INCR >= N
       MULTIPLY INCR BY 3
       ADD 1 TO INCR
       MULTIPLY INCR BY 9 GIVING T-INCR
       ADD 4 TO T-INCR.
     END-PERFORM.
     PERFORM UNTIL INCR <= 0
        ADD 1 TO INCR GIVING I
        PERFORM VARYING I FROM I BY 1
                          UNTIL I > N
            SUBTRACT INCR FROM I GIVING J
            PERFORM UNTIL J <= 0
                ADD J TO INCR GIVING K
                IF THE-KEY ( J ) > THE-KEY ( K ) THEN
*                  SWAP THE ITEMS
                   MOVE THE-ITEMS ( J ) TO TEMP-ITEM
                   MOVE THE-ITEMS ( K ) TO THE-ITEMS ( J )
                   MOVE TEMP-ITEM TO THE-ITEMS ( K )
                   SUBTRACT INCR FROM J
                ELSE
                   MOVE 0 TO J
                END-IF
            END-PERFORM
        END-PERFORM
        SUBTRACT 1 FROM INCR
        DIVIDE 3 INTO INCR
     END-PERFORM.
```

12.13.11 OPEN-FEEDBACK Copy Book (OPENFB)

```
*----------------------------------------------------------------*
*  OPEN FEEDBACK INFORMATION                                     *
*----------------------------------------------------------------*
 01  OFB-INFORMATION.
     05  OFB-DEVTYPE       PIC X(2).
         88  OFB-DEVICE-FILE     VALUE ″DS″.
         88  OFB-DATA-BASE-FILE  VALUE ″DB″.
         88  OFB-SPOOL-FILE      VALUE ″SP″.
     05  OFB-FILE-LIB.
         10  OFB-FILE-NAME     PIC X(10).
         10  OFB-LIBRARY-NAME  PIC X(10).
*        FOR SPOOL FILE WILL BE ″*N″
     05  OFB-SPOOL-NAME    PIC X(10).
     05  OFB-SPOOL-LIB     PIC X(10).
     05  OFB-SPOOL-NBR     PIC S9(4) BINARY.
     05  OFB-REC-LENGTH    PIC S9(4) BINARY.
     05  FILLER            PIC X(2).
     05  OFB-MBR-NAME      PIC X(10).
     05  FILLER            PIC X(8).
     05  OFB-FILE-TYPE     PIC S9(4) BINARY.
         88  OFB-DISPLAY     VALUE 1.
         88  OFB-PRINTER     VALUE 2.
         88  OFB-DISKETTE    VALUE 4.
         88  OFB-TAPE        VALUE 5.
         88  OFB-SAVE        VALUE 9.
         88  OFB-DDM         VALUE 10.
         88  OFB-ICF         VALUE 11.
     05  FILLER            PIC X(3).
     05  OFB-NBR-LINES     PIC S9(4) BINARY.
     05  OFB-NBR-COLS      PIC S9(4) BINARY.
     05  OFB-NBR-RECS      PIC S9(9) BINARY.
     05  OFB-ACCESS-TYPE   PIC X(2).
         88  OFB-KEYED-UNIQUE VALUE ”KU”.
         88  OFB-KEYED-FIFO   VALUE ”KF”.
         88  OFB-KEYED-LIFO   VALUE ”KL”.
         88  OFB-ARRIVAL-SEQ  VALUE ”AR”.
     05  OFB-DUPLICATES    PIC X.
*        D = DUPS   U = UNIQUE
     05  OFB-SOURCE-FILE   PIC X.
*        Y = SOURCE
     05  OFB-UFCB-PARMS    PIC X(10).
     05  OFB-OVR-PARMS     PIC X(10).
     05  OFB-VOLLAB-OFFSET PIC S9(4) BINARY.
     05  OFB-MAX-BLKRCDS   PIC S9(4) BINARY.
     05  OFB-PRT-OVERFLOW  PIC S9(4) BINARY.
     05  OFB-BLK-INCREMENT PIC S9(4) BINARY.
     05  OFB-EXTENSION-OFFSET PIC S9(9) BINARY.
     05  OFB-MISC-FLAGS    PIC X.
*        BIT 1 = ACCEPT INPUT BEING USED
*        BIT 2 = FILE OPENED SHARABLE
*        BIT 3 = FILE UNDER COMMITMENT CONTROL
*        BIT 4 = ALL RECORDS ACCESSED ARE LOCKED
*        BIT 5 = MEMBER IS A LOGICAL MEMBER
*        BIT 6 = FILE CREATED WITH DDS
*        BIT 7 = FILE IS DBCS CAPABLE
*        BIT 8 = END OF FILE DELAY PROCESSING BEING DONE
     05  OFB-PGM-DEVICE    PIC X(10).
     05  OFB-OPEN-COUNT    PIC S9(4) BINARY.
     05  OFB-LOWEST-IND    PIC S9(4) BINARY.
     05  OFB-NBR-MBRS-OPEN PIC S9(4) BINARY.
     05  OFB-MISC-FLAGS2   PIC X.
     05  FILLER            PIC X(13).
     05  OFB-NBR-PGM-DEV   PIC S9(4) BINARY.
*  PROGRAM DEVICE LIST FOLLOWS HERE
```

12.13.12 I-O-FEEDBACK Copy Book (IOFB)

```
*---------------------------------------------------------------*
*  I-O  FEEDBACK INFORMATION                                    *
*       ACCEPT INTO THE IOF AREA THAT MATCHES FILE TYPE         *
*---------------------------------------------------------------*
*
*  DATABASE I/O FEEDBACK INFORMATION
*
 01  IOF-DB.
     05  FILLER               PIC X(144).
     05  IOF-DB-TEMPLATE-SIZE PIC S9(9) BINARY.
     05  IOF-DB-JOINFILE      PIC X(4).
     05  FILLER               PIC X(2).
     05  IOF-DB-LOCKED-RECS   PIC S9(4) BINARY.
     05  FILLER               PIC X(7).
     05  IOF-DB-REC-DELETED   PIC X.
*        HEX 00 - NOT DELETED, HEX 10 - REC DELETED
     05  IOF-DB-NBR-KEY-FIELDS PIC S9(4) BINARY.
     05  FILLER               PIC X(4).
     05  IOF-DB-KEY-LENGTH    PIC S9(4) BINARY.
     05  IOF-DB-MBR-NUMBER    PIC S9(4) BINARY.
     05  IOF-DB-REL-NUMBER    PIC S9(9) BINARY.
     05  IOF-DB-KEY           PIC X(121).
*
```

```
*  DISPLAY AND ICF I/O FEEDBACK INFORMATION
*
 01  IOF-DSP-AND-ICF REDEFINES IOF-DB.
     05  FILLER            PIC X(144).
     05  IOF-FLAG-BITS     PIC X(2).
*        BIT 1 - GET CANCEL INDICATOR
*        BIT 2 - DATA RETURNED INDICATOR
*        BIT 3 - COMMAND KEY INDICATOR
*        BITS 4-16 RESERVED
     05  IOF-DSP-AID-BYTE      PIC X.
     05  IOF-DSP-CURSOR-LINE   PIC X.
     05  IOF-DSP-CURSOR-COL    PIC X.
     05  IOF-ICF-DTALEN    PIC S9(9) BINARY.
     05  IOF-SFL-REC-NBR   PIC S9(4) BINARY.
     05  IOF-SLF-LOW-NBR   PIC S9(4) BINARY.
     05  IOF-SLF-REC-COUNT PIC S9(4) BINARY.
     05  FILLER            PIC X(19).
     05  IOF-DSP-MAJOR-MINOR.
         10  IOF-DSP-MAJOR-RC  PIC X(2).
             88  IOF-DSP-SUCCESS      VALUE ″00″.
             88  IOF-DSP-CANCELING    VALUE ″02″.
             88  IOF-DSP-NO-DATA      VALUE ″03″.
             88  IOF-DSP-OUTPUT-ERR   VALUE ″04″.
             88  IOF-DSP-ACQUIRED     VALUE ″08″.
             88  IOF-DSP-SYS-ERR      VALUE ″80″.
             88  IOF-DSP-DEV-ERR      VALUE ″81″.
             88  IOF-DSP-OPEN-ERR     VALUE ″82″.
             88  IOF-DSP-SESSION-ERR  VALUE ″83″.
         10  IOF-DSP-MINOR-RC  PIC X(2).
     05  IOF-DSP-RSP-RC    PIC X(8).
     05  IOF-DSP-SAFE-IND  PIC X.
     05  FILLER            PIC X.
     05  IOF-ICF-REQ-WRITE PIC X.
     05  IOF-ICF-RMT-FORMAT PIC X(10).
     05  IOF-ICF-MODE-NAME  PIC X(8).
     05  FILLER             PIC X(9).
*
*  PRINTER I/O FEEDBACK INFORMATION
*
 01  IOF-PRINTER  REDEFINES IOF-DB.
     05  FILLER            PIC X(144).
     05  IOF-PRT-CUR-LINE  PIC S9(4) BINARY.
     05  IOF-PRT-CUR-PAGE  PIC S9(9) BINARY.
     05  FILLER            PIC X(28).
     05  IOF-PRT-MAJOR-MINOR.
         10  IOF-PRT-MAJOR-RC  PIC X(2).
             88  IOF-PRT-SUCCESS      VALUE ″00″.
             88  IOF-PRT-CANCELING    VALUE ″02″.
             88  IOF-PRT-SYS-ERROR    VALUE ″80″.
             88  IOF-PRT-SES-ERROR    VALUE ″81″.
             88  IOF-PRT-ACQUIRE-FAIL VALUE ″82″.
             88  IOF-PRT-SESSION-ERR  VALUE ″83″.
         10  IOF-PRT-MINOR-RC  PIC X(2).
*
*  COMMON FEEDBACK INFORMATION FOR ALL FILES
*
 01  IOF-COMMON REDEFINES IOF-DB.
     05  IOF-DEV-OFFSET    PIC S9(4) BINARY.
     05  IOF-OUTPUT-COUNT  PIC S9(9) BINARY.
```

```
           05  IOF-INPUT-COUNT   PIC S9(9) BINARY.
           05  IOF-OUTIN-COUNT   PIC S9(9) BINARY.
           05  IOF-OTHER-COUNT   PIC S9(9) BINARY.
           05  FILLER            PIC X.
           05  IOF-CUR-OP        PIC X.
           05  IOF-FORMAT-NAME   PIC X(10).
           05  IOF-DEV-CLASS     PIC X(2).
           05  IOF-DEV-NAME      PIC X(10).
           05  IOF-RECORD-LEN    PIC S9(9) BINARY.
           05  FILLER            PIC X(80).
           05  IOF-NBR-RECORDS   PIC S9(4) BINARY.
           05  FILLER            PIC X(4).
           05  IOF-TAPE-BLOCKS   PIC S9(9) BINARY.
           05  FILLER            PIC X(8).
```

12.13.13 ATTRIBUTE-DATA Copy Book (IOATTR)

```
      *----------------------------------------------------------------*
      *  PROGRAM DEVICE ATTRIBUTES                                     *
      *----------------------------------------------------------------*
       01  ATR-INFORMATION.
           05  ATR-DEV-NAME        PIC X(10).
           05  FILLER              PIC X(50).
           05  ATR-DESC-NAME       PIC X(10).
           05  ATR-DEV-CLASS       PIC X.
           05  ATR-DEV-TYPE        PIC X.
           05  ATR-NBR-ROWS        PIC S9(4) BINARY.
           05  ATR-NBR-COLS        PIC S9(4) BINARY.
           05  ATR-BIT-FLAGS       PIC X.
      *        BIT 1 = DISPLAY CAPABLE OF FLASHING
      *        BIT 2 = REMOTE DEVICE
      *        BIT 3 = DEVICE ACQUIRED
      *        BIT 4 = DEVICE INVITED
      *        BIT 5 = DATA AVAILABLE
      *        BIT 6 = TRANSACTION STARTED
      *        BIT 7 = *REQUESTER DEVICE
      *        BIT 8 = RESERVED
           05  FILLER              PIC X.
           05  ATR-SYNCH-LVL       PIC X.
```

12.14 AS/400 ILE C/400 Tips and Techniques

V3R6 ILE C/400 has made improvements to compile-time and run-time performance as well as providing new mechanisms to allow programmers the ability to improve the performance of their applications. Programs developed in previous releases of ILE on the AS/400 system need to be re-translated or re-compiled in order for them to take advantage of V3R6 performance improvements.

12.14.1 ILE C/400 Function In-Lining

V3R6 C/400 provides a new inline capability that allows small functions to be ″in-lined″ or imbedded at the point of call, which reduces the number of internal calls performed at run time. For highly used functions, this capability can significantly improve application performance.

12.14.2 ILE C/400 Exception Handling

V3R6 C/400 provides a new exception handling option that can reduce the overhead required to handle an exception. An action can be specified directly rather than having an exception handling function called to perform the action.

12.14.3 ILE C/400 Heap Management

V3R6 C/400 has changed heap management and program name processing.

Heap pages are brought into main storage in 4K blocks (8 pages); this is the standard AS/400 page size.

In V3R6, C/400 now saves the program name after the first call to the program, so if subsequent calls are made to the program in the same activation group, the overhead to get the program name is eliminated. (ILE Call Bound program calls are still faster than this technique.)

12.14.4 ILE C/400 MUTual EXclusion (MUTEX) Instruction

V3R6 C/400 provides a new set of MUTEX MI instructions as built-ins to provide the UNIX-based programmer a fast mechanism for synchronizing access to a shared object. This is similar to a ″locking function″. DSPJOB provides a ″display mutexes″ option so you can see the mutex status.

12.14.5 ILE C/400 Scan Wild Card (SCANWC) Function

V3R6 C/400 provides a faster implementation for the SCANWC( ) function.

12.14.6 ILE C/400 Stream I/O Improvement

V3R1 C/400 provided a new implementation of stream I/O support available under the CPA Toolkit (UNIX-based support) technology on preview support. For V3R6, the support for environment variables shared memory and semaphores was removed from the CPA toolkit and is now part of OS/400. The support for threads remains part of the CPA Toolkit.

Because the support for the environment variables, shared memory, and semaphores is now part of OS/400, the compiled object code from an application that uses these APIs in V3R1 is not compatible with V3R6.

Therefore, you must compile the application again, also the source code for any applications that use environment variables, shared memory, or semaphores must be updated and recompiled again.

Threads continue to be part of the CPA toolkit. However, the source code for any applications that uses threads must also be updated and compiled again.

12.14.7 ILE C/400 Variable and Data Type Usage Tip

You should avoid using a volatile qualifier as this prohibits optimization.

Avoid using bit-fields, since it takes more time to access bit-fields than other data types such as short and integer. Avoid using static and global variables as they are initialized whether or not you explicitly initialize them. The performance improvement here is only for the application or the activation group startup.

Use the register storage class for a variable that is frequently used.

However, do not over use placing variables into hardware registers.

Avoid using packed structures as there is some overhead when accessing fields in a packed structure.

Use the #pragma strings (read only) directive to reduce the amount of static storage required to store string literals. If identical strings appear in the same program, storage is allocated once for the string.

12.14.8 ILE C/400 Exception Handling Recommendation

You should reduce the number of exceptions generated because exception handling is expensive for ILE C/400 programs. Turn off C2M messages during record I/O. This controls the messages sent to your application when it detects certain types of errors (such as record truncation). Use the *#pragma nosigtrunc directive* to turn off overflow exceptions from packed decimal arithmetic operations.

Use a direct monitor handler instead of a signal handler. Avoid percolating exceptions. Try to handle the exception in the place it occurs.

12.14.9 ILE C/400 File I/O Interface Recommendations

Using record I/O functions instead of stream I/O functions can greatly improve I/O performance. Instead of accessing one byte at a time, record I/O functions access one record at a time. This recommendation may be modified based on experience with new stream I/O interfaces available in the CPA Toolkit previously discussed within this C/400 section.

You can improve record I/O performance by blocking records. When blocking is specified, the first read causes an entire block of records to be placed into a buffer. Subsequent read operations return a record from the buffer until the buffer is empty. At that time, the next block is fetched. You can improve I/O performance of your ILE C/400 programs by performing read and write operations directly to and from the system buffer without the need for an application-defined buffer. Directly manipulating the system buffer provides a performance improvement when you process long records. It also provides a significant performance improvement when you use Intersystem Communication Function (ICF) files.

For all application programs on the AS/400 system, you can improve the performance of your application by not opening the same file more than once. You can allocate the file pointers as global (external) variables, opening the file once, and not closing the file until the end of the application.

Since stream I/O functions cause many function calls, reducing their use in your application improves performance.

To improve your performance, you should use physical files instead of source physical files for your data. When a source physical file is used for stream I/O, the first 12 bytes of each record are not visible to your application. They are used to store the record number and update time. These 12 bytes are an extra load that the ILE C/400 stream I/O function must manipulate.

12.14.10 ILE C/400 Pointer Recommendations

You should avoid using open pointers as they prohibit optimization. Since a pointer takes up 16 bytes of space, pointer comparisons in ILE C/400 language are less efficient than comparisons using other data types. You may want to replace pointer comparisons with comparisons using other data types, such a INTeger.

However, using pointers is more efficient than using subscripts to access multi-dimensional arrays, as this saves a number of subscript arithmetic operations that often involve multiplications.

12.14.11 ILE C/400 Program Storage Space Recommendations

You can improve the performance of an application by reducing the space required for an ILE C/400 application. Reducing the space requirement helps reduce page faults and segment faults. It also helps reduce the number of allocations beyond the size of the fast heap.

The first request for dynamic storage within an activation group results in the creation of a default heap from which the storage allocation takes place. On the first call within an activation group to the *malloc* or *calloc* functions, ILE C/400 creates a fast heap of a specified size up to 16MB. Storage requests from this heap are reasonably fast. Storage allocations beyond the size of this heap are much slower. The default size of the fast heap is 64K. It is recommended that you change this size to a size large enough to handle the dynamic storage requirements of your application. This way, your dynamic storage requests are always satisfied using the fast heap, and this can have a noticeable performance improvement.

Due to the new increase in the default page size of 4KB, Effective Address Overflows (EAO) are not a major concern (and are no longer reported in the Performance Tools Component Report).

The Performance Tools/400 Manager Feature reports exception counts. The Advisor function or tables shown in Appendix A, “Guidelines for Interpreting Performance Data” on page 383 can be used to assess CPU impact based on the AS/400 Relative Processor Performance value.

Chapter 13. AS/400 Client/Server and File Serving Performance

This chapter covers in broad terms the performance of the AS/400 system when used as a server. Two categories are covered: using the AS/400 system as a generic server for client/server applications, and using the AS/400 system specifically for PC file serving.

13.1 Client Server Performance Tips and Considerations

This section contains a summary of performance tips and performance considerations for *some* of the client/server software products available on the AS/400 system. References are made to other documentation for a more complete discussion of performance considerations for a specific product. Some product positioning information is presented but a complete ″function versus performance″ positioning evaluation for each client/server application is beyond the scope of this redbook.

Client/server products discussed in this section include:

- LAN Server/400 and Integrated PC Server
- Client Access/400
- ADSTAR Distributed Storage Manager/400 (ADSM/400)
- LANRES/400
- Ultimedia System Facility/400 Facilities/400
- ImagePlus Workfolder Application Facility/400
- DataPropagator Relational/400
- OptiConnect/400

Use the information in this redbook for performance management information, but refer to the *Performance Capabilities Reference*, ZC41-8166 (for V3R2) or ZC41-0807 (for V3R7), for detailed performance tips and test results of these and other client/server applications.

Before discussing the specific products, there are two topics that apply to all client/server environments and they are discussed first:

- Importance of data and program processing placement
- Definition of client/server application types

13.1.1 Importance of Data and Program Processing Placement

Regardless of the client/server application type, the following items must always be considered within a client/server environment:

- Group the most frequently changed data together for best management (security and performance):

  Infrequently changed data can be downloaded to the client at the start of a session. This minimizes repetitive transmission of this data across a communication link for every transaction.

  Data changed infrequently can be placed in separate ″storage areas″ from data that is changed frequently. This expedites the backup process by

″saving″ only the changed data as a separate data storage area rather than as individual files, if possible.

- Consider using a separate storage pool and expert cache to improve performance on the AS/400 system for a client/server function.

  As shipped, many of the client/server applications use the *BASE storage pool. *BASE is used by many other applications and some OS/400 system functions. Consider using a separate storage pool for a specific function. Consider also using expert cache for that storage pool. Test the environment both with expert cache and without expert cache to evaluate any performance impact.

  See subsequent *work management* topics for the client/server applications included in this redbook for information on assigning storage pools and run priorities for these applications.

- Use the fastest communication links and minimize connecting bridges or routers:

  This requirement is obvious but often the customer does not realize the amount of traffic over a communication link or that bridges or routers between fast links can become saturated and act as bottlenecks during peak transmission periods. This is especially important if multiple protocols are being used.

- Use the fastest client workstation:

  Any client that must perform some of the processing of a client/server function should be the fastest possible. If not, there can be periods of time where the server has to wait for a response from the client. In some environments, this may slow the server′s capability to perform requests received from other clients.

- Performance tune the client workstation:

  When setting up the client/server application environment, tuning the AS/400 system and the client must be completed to get the best possible performance. Experience indicates that not only is a fast client processor required, but specifying the ″best″ client parameter values is critical to satisfaction with performance.

  Subsequent client/server application sections in this redbook contain discussions of client parameter values that improve performance.

13.1.2 Client/Server Application Types

Client/server environments generally include one or more client workstations connected to at least one server. The amount of processing performed for a particular end-user function can be performed mostly on the client, mostly on the server, or a shared processing on both the client and the server. This requires the persons responsible for performance management to understand the application implementation in order to assess ″workstation″ resources utilized and manage these utilizations against a perceived response time or throughput, such as ″bytes-per-second″ sent or received.

In this redbook, the following terms and definitions are used for the different *client/server application types*:

- Server:

The server is considered to be the AS/400 system or the Integrated PC File Server attached to the AS/400 system when LAN Server/400 is being used.

- Client:

  The client workstation is considered to be a personal computer or system or workstation with software defined to be *dependent on the server for the function being performed*. In other words, the software is defined to play the client role.

- File serving:

  File serving represents the server acting as a *network drive* (″virtual disk″) for the client. The network drive may contain data or programs or procedures.

- Database serving:

  The server contains data that the client knows and understands can be on a remote server. Client program interfaces are different between file server and database server accesses. Often only subsets of the server files are processed by the client.

  APIs such as Open Database Connectivity (ODBC) and Remote SQL are used in this application type.

- Print serving:

  Printer serving includes one or both of the following:

  - Server (″host″) printers are used by the client as if they were attached to the client.
  - Client printers are used by the server as if they were attached to the server (host).

- Application serving:

  An application function is shared by processing on both the server and the client. The amount of processing done on the client versus the server is dependent on application design and in consideration of how much other application processing the server (host) may also be performing during the specific client/server transaction.

  This application implementation offers the best chance at achieving maximum performance but requires significant programming skills and debugging time to reach that maximum performance.

- Administration:

  In this redbook, *administration* includes the functions typically assigned to a server for defining users, data and program resources, and the authorization of client users to the data and program resources.

  For various licensed programs, administration also includes:

  - The amount of storage space that can be used by a client
  - Save and restore functions for the server and the client data and program resources
  - Software license and resource management

13.2 AS/400 File Serving Performance

In general, V3R6 file serving performance for Client Access/400, and for LAN Server/400 with Integrated PC Server file serving is equivalent to V3R1 when comparing systems of equal Relative Performance Rating (RPR). For information on RPRs, please refer to Appendix A, “Guidelines for Interpreting Performance Data” on page 383.

This chapter focuses on the following topics:

- File serving performance positioning
- Client Access/400 File serving with the Integrated File System (IFS) File Server/400
- Integrated PC Server and LAN Server/400 File serving
- Multimedia file serving

For information on the AS/400 system versus Microsoft′s SNA Server Gateway, refer to section Appendix I, “Related Publications” on page 431 for directions on how to access a presentation on this topic.

For performance of various communication and connectivity methods, see Chapter 12, “Design and Coding Tips” on page 229

13.3 File Serving Performance Positioning

With the unique requirements in the area of PC file serving, it is important to provide an environment to meet the needs of the customer. For PC file serving on the AS/400 system. there are basically two options: LAN Server/400 on the Integrated PC server and Client Access/400 (NetWare on the Integrated PC server is available with OS/400 V3R7). If your requirement is for ***high performance*** file serving from client workstations running DOS, Microsoft Windows, or OS/2 attached through LAN to an AS/400 system, LAN Server/400 and the Integrated PC Server is the best solution. LAN Server/400 provides competitive PC file serving performance comparable to the leading PC servers, and requires dramatically less AS/400 CPU resource than using Client Access/400 and the Integrated File System for file serving.

Where performance is not a key requirement in PC file serving, such as in the areas of casual file serving or client administration, file serving provided by Client Access/400 is sufficient. The strength of Client Access/400 is in providing seamless integration of the PC desktop not only in the area of file serving, but also in areas such as database serving, print serving, AS/400 application access, and 5250 emulation.

When you require both high performance file serving and access to all the PC desktop services provided by Client Access/400, LAN Server/400 and Client Access/400 can be used concurrently on the client. The LAN Server/400 requester provides access to all data maintained by LAN Server/400. Client Access/400 provides access to other AS/400 resources such as the AS/400 database, byte stream data in the Integrated File System (IFS), and printers.

IFS integrates the existing AS/400 file systems (QSYS.LIB - libraries/objects, and QDLS - folders/documents/shared folders) into a single hierarchical name space together with the following file systems:

- ″Root″: OS/2 and DOS compatible
- QOpenSys: POSIX, XPG, and UNIX** compatible
- QLANSrv: LAN Server/400 data - OS/2 LAN Manager compatible
- QNETWARE: Netware server data (V3R7 only)

13.3.1 File Serving Workloads and Configurations

The following workload environments and client/server configurations are used in this chapter to compare file serving performance:

- **BAPCo5 Workload Description**:

  The BAPCo5 (Business Application Performance Corporation) workload represents a client/server environment in which PC users run the following five commercially available applications and access programs, batch files, and data files residing on the server. Some of the functions utilized by each application are listed.

  - **Harvard Graphics**: macros, create org chart, 3-D pie chart, import Lotus and Excel spread sheets, plot, create text chart, slide show, print
  - **Paradox**: define databases in the file system, selective retrieval, report generation, print
  - **WordPerfect**: small and large documents, scroll, help feature, tables, headings, search and replace, edit
  - **Excel**: bivariant normal distribution, print product forecasting, create tables, cut/paste, data entry, 2-D and 3-D graphs, format, print tax forms, load forms, enter data, scroll, links, print preview, print
  - **CCMail**: electronic mail, selective retrieval, edit, send response, print

  The preceding applications and end-user functions were selected based on research on which applications and functions are most prevalent.

  The applications and the files are maintained on a networked server and are accessed from client machines running Win-OS/2. Each client machine runs an automated script (beginning with a different application) and runs one application at a time until the entire set of applications has been run. The automated scripts run with zero think times and thus generate file serving requests that are representative of many clients. Print output is held in a print queue on each client.

  Note: The BAPCo5 workload is a modification of the BAPCo workload described in the *Performance Capabilities Reference for PowerPC Technology*, ZC41-0607.

13.4 Client Access/400 File Serving Performance

OS/400 V3R6 supports the Windows 3.1 native client (16-bit), the Optimized OS/2 client (32-bit), and the Windows95 Client (32-bit). Each is designed to provide integration with their respective desktop operating system, and also take full advantage of the AS/400 Integrated File System. The Windows 3.1 Client provides native Windows communication support. Only adapter drivers, such as the LAN Support Program, run as DOS TSR (terminate and stay resident) programs. Less real memory is required as a result of this change to eliminate most TSR programs. The Windows95 Client supports connection to the AS/400 system using TCP/IP protocol as well as APPN.

For all three previously mentioned clients, the Root file system offers the best file serving performance. Therefore, as users migrate to using these clients from the existing DOS clients and the original OS/2 client (16 bit), it is recommended to also move their data into the new Root file system from the QDLS file system to get the best performance. However, this is not always possible. In environments where new clients, such as Windows 3.1, and original clients, such as DOS extended, need to access the same data, the data needs to be stored in the QDLS (shared folders) file system.

13.4.1 Performance Tips/Techniques for Client Access/400 File Serving

Throughout this section, ″IFS File Serving″ refers to the functions formerly known as ″Shared Folders″.

13.4.1.1 Setup and Configuration Tips and Techniques

All Clients:

1. The following information should be considered when choosing the size of the cache for CA/400 clients:

   - Use a small cache (256 kilobytes - 1MB) for applications accessing data sequentially in small amounts (or if you do not have much available PC memory).
   - Use a medium to large cache (500KB or larger) for applications accessing data randomly.
   - Use a medium to large cache (500KB or larger) for applications accessing data both randomly and sequentially.
   - The required cache size varies with the application. Creating a cache larger than is necessary may not further improve the performance of that application. However, if the caching is effectively reducing communication interaction with the AS/400 system, the load on the AS/400 system processor is reduced.
   - If you typically run most of your applications once a day and use the data associated with those applications only once, the cache size to choose is the maximum cache size determined for those applications.
   - If you typically run some or all of your applications more than once and use the data associated with those applications more than once, the cache size to choose is the sum of the cache sizes determined for those applications.

     Note: Increased cache sizes increase the amount of memory used on the PC. If there is already memory usage problems on the PC, increasing the cache size for CA/400 makes the problem worse.

2. Use a large frame size if possible. Significant performance improvement may be achieved for Windows 3.1 clients when using large frame size.

DOS Clients:

1. The following information should be considered when choosing the size of the cache for DOS clients:

   - Client Access/400 supplies a tool called GETSTAT that can help you tune your PC memory cache size. The tool is in the AS/400 folder QIWSTOOL. Run the program IWSTOOL that is also in the QIWSTOOL folder and select GETSTAT to download to your fixed disk or a diskette. GETSTAT tells you the memory used by your IFS File Serving programs and buffers. It also gives you an idea of how effectively your cache is working to limit data transfer with the AS/400 system.

Note: GETSTAT only works with the DOS Extended client.

- With DOS Extended, use a minimum cache of 128 kilobytes.
- The value specified on the MCAC, MCAE, and MCAX identifiers is used to create both the IFS File Serving cache and a cache table to keep track of things within the cache.

2. If you are using DOS Extended (XMS), space for the cache as specified on the MCAX parameter is allocated from extended memory. Using XMS does not permit the allocation of cache and the associated cache table in conventional memory.

3. If you are using DOS 5.0 or later, and have the following statement in your CONFIG.SYS file, it is possible for DOS Extended IFS File Serving code to only use 96 bytes of your conventional memory.

```
dos=high,umb
```

4. There are two connectivity types that do support the placement of the communications buffer in extended memory: SDLC and ASYNC. In these cases, the FEMU is ignored and the buffer placed into extended memory. When possible, specify this in conventional memory for performance advantages.

5. The size of the communications buffer can have an effect on the performance of IFS file serving. The size is specified with the CBSZ identifier in the configuration file (CONFIG.PCS). Note that the size of the communications buffer can be changed only if it is located in conventional memory with the exception as noted earlier for the two connectivity types that support extended memory (ASYNC and SDLC).

   - If enough memory is available on the PC, increasing the size of the communications buffer can improve response times.
   - If the PC is having problems with memory usage, use the default communications buffer size (8K).

OS/2 Clients:

1. The LAN Server/400, the Integrated PC Server, the IBM OS/2 LAN Server, or the PC LAN program may be better file-serving alternatives if heavy file I/O is the only function to be performed. Client Access/400 should be considered when many of the required functions are not available with other file servers. These functions include host data transfer, host integration, 5250 emulation, and remote system access.

   Since Client Access/400 can reside simultaneously with a PC file server (that is, LAN Server/400, OS/2 LAN Server, or PC LAN Program) in the same workstation, it may be desirable to install more than one server on the LAN. For example, the PC LAN Program can be used to provide the performance needed for program loading from a PC server while IFS File Serving Function can be used for storing data for remote PCs and for data exchange between PCs and the host system.

   Another alternative is the NetWare for SAA product that supports NetWare serving from a PC server. It also supports all the Client Access/400 products from an AS/400 system attached to the NetWare PC server.

13.4.1.2 Application Tips and Techniques

In general, PC hard disks typically provide better performance than the IFS File Serving function. However, there are techniques to lessen performance differences. The following techniques may be used:

1. When appropriate, store some PC programs/files on the PC hard disk.

2. Copy PC files to a PC RAM disk or hard disk, use the files from the RAM disk/hard disk, and copy the changed files back to the AS/400 system when the work is complete.

3. Minimize the number of times a file is uploaded or downloaded.

4. A PC application should use an appropriate create or open operation instead of searching for the existence of a file. Searching for a file requires significant time and resources compared with other operations.

5. PC applications should be designed to open a file once, perform all necessary operations, and close the file.

6. Use ″write with verification″ only when absolutely necessary. When verification is used, write operations are not buffered by the File Server/400.

7. When reading data from a file, read from beginning to end and not randomly. The File Server/400 buffers are used more efficiently when files are accessed sequentially from beginning to end.

8. Design your PC applications so that small files are read into memory. Memory accesses are always faster than accessing the disk or data on the AS/400 system.

9. Increase the LANMAXOUT value in the AS/400 controller description for clients attached by a TRLAN. Changing the default value from two to seven provided significant performance improvement during testing in a lab environment. Changing the default value to six may yield similar improvement.

10. Backing up files to a folder can take significant AS/400 resources. To avoid performance impacts to other AS/400 applications, consider the following options:

    - It is possible to use a packaging utility such as PKZIP to create a single large file that contains images of multiple files. The single large file can be saved for back-up purposes. This can eliminate hundreds of creations of files, and can literally make the time to back up hundreds of times faster.
    - Back up large numbers of files to folders when system activity is low.
    - Use the Change Job (CHGJOB) command to decrease the run-time priority of the job doing the backup so that system resources are not tied up.

11. For the new clients, the Root or QOpenSys file systems offer best file serving performance.

12. Avoid doing many open/delete operations from the same Root or QOpenSys directory. Delete causes the name cache for the specified directory to be invalidated and results in poor performance. Try to keep directories that have frequent deletions (for example, temp files) separate from directories containing files that do not frequently get deleted.

13. Avoid any administration (create, delete, update operations) of authorization lists during peak IFS usage. This improves the efficiency of internal caches.

14. Avoid using supplemental group profiles. Use the default value of *NONE for the SUPGPPRF parameter in the user profile. Best performance is achieved for users that do not have supplemental group profiles.

15. The Root or QOpenSys file system supports hard links and symbolic links. Hard links can improve performance by reducing the number of directory lookups. However, having many hard links to the same file can result in poor performance. Best results are achieved when you use a small number of hard links for key files that are commonly used.

16. For Optimized OS/2 clients, use ″current working directory″. This can be done by assigning a drive to the working directory or use the CD command. Using current working directory can reduce the number of directory lookups and improve performance.

17. Avoid doing many MKDIR and RMVDIR operations when using the Root or QOpenSys file system.

18. Avoid doing many create and delete operations when using the QDLS file system.

13.4.1.3 File Server/400 Tuning

The following tips can be used to tune File Server/400 (IFS File Server on the AS/400 system). Client Access/400 clients use this server when accessing network drives. There is always the risk of giving too much of any one resource to a particular job, so consider the other jobs on the AS/400 system and their importance before shifting resources to the File Server/400.

As with other jobs, run priority and time slice can be adjusted for File Server/400 jobs. For File Server/400, this can be accomplished by changing job class QPWFSERVER.

Performance of File Server/400 also benefits from having more memory available. One option to provide File Server/400 jobs with adequate memory is to associate the QSERVER subsystem with its own pool and set aside the appropriate amount of memory for that pool.

Multiple clients executing the BAPCo workload were used to determine the optimal amount of memory needed per client. It was found that in the range of 1.5MB to 2.0MB per user, client response times ″leveled off″ such that more memory did not significantly improve response times. As the response times leveled off, the paging in this pool was still higher than recommended, but since the DASD arms were at relatively low utilization, the additional IO did not significantly affect response time. However, as additional memory was added to the pool beyond 2.0MB per user, the paging and faulting for that pool continued to decrease significantly until about 4.0MB was available for each client (QPWFSERV file server job to be more exact).

13.5 LAN Server/400 and Integrated PC Server File Serving Performance

LAN Server/400 is an AS/400 V3R6 licensed program that works with the AS/400 Integrated PC Server to provide high performance file serving to PC LAN clients.

LAN Server/400 implements OS/2 LAN Server technology on the AS/400 system and allows the AS/400 system to provide services similar to those of an OS/2 LAN Server. PC clients can access files stored on the AS/400 system by using the IBM LAN Requester program, which is the same software used to request service from an OS/2 LAN Server. From the perspective of PC clients, the AS/400 system and OS/2 LAN Server are both servers that can store data and programs. PC LAN clients running DOS, OS/2, or Microsoft Windows are all able to request services.

LAN Server/400 and Integrated PC Server provide high performance file serving at a level not previously available on the AS/400 system. Response time and throughput characteristics of file serving with an Integrated PC Server are consistently comparable with PC LAN file servers such as OS/2 LAN Server and Novell NetWare running on equivalent hardware. This is true for a small or large number of users.

The remainder of this section gives performance information for several scenarios using LAN Server/400 and Integrated PC Server. For a thorough understanding of LAN Server/400 features and function, refer to *LAN Server/400 - A Guide to Using the AS/400 as a File Server*, GG24-4378. Another useful reference document is *LAN Server for OS/400 Administration V3R1, V3R6*, SC41-3423-01.

QLANSrv is a file system that is present on systems that have the LAN Server/400 licensed program installed. QLANSrv provides a mechanism for host applications and the OS/400 Integrated File System (IFS) file server (File Server/400) to access data that is in the LAN Server/400 file system.

Note: Host applications or PC clients (using File Server/400) that access files through QLANSrv do not achieve the same level of performance that PC clients using LAN Requester applications can achieve.

For information on Saving/Restoring LAN Server/400 files and storage spaces, see Section 13.5.4, “LAN Server/400 Save/Restore Performance” on page 347.

LAN Server/400 and Integrated PC Server provide equivalent performance with a 2617 or 2619 IOP when used as a LAN communications controller. The Integrated PC Server has greater capacity for large transfer scenarios (achieving rates over 14 Mbps).

Refer to the *AS/400 Advanced Series handbook*, GA19-5486-14, for additional information when comparing Integrated PC Server (6506 IOP) with 2617 and 2619 IOPs.

13.5.1 LAN Server/400 and Integrated PC Server Sizing Guidelines

LAN Server/400 Cache:

The LAN Server/400 product is configured to recognize the amount of memory installed on the Integrated PC Server and to allocate cache depending on the amount of memory installed and the type of card used (one or two port). There are no options available on LAN Server/400 to change these values.

See Table 29 on page 338 for more details on cache memory size.

13.5.1.1 Estimating the Number of Clients Supported by an Integrated PC Server

The number of clients that an Integrated PC Server can support when used as a file server depends on the amount of memory available for the HPFS cache on the Integrated PC Server, the rate of requests from the clients, client hardware configurations, LAN utilization, and other factors. It is difficult to generalize with so many variables.

Table 27 shows guidelines for the number of users supported. For casual file serving, the number of supported users is larger. For heavy file serving (for example, multimedia continuous medium-quality video that delivers data to each client at a rate of 150KB per second), the number of supported users is lower, as resources such as the LAN become a bottleneck.

Table 27. AS/400 Integrated PC Server - Getting Started

Guidelines for Getting Started	
Integrated PC Server Memory Size	**Number of Users**
16MB	1 - 20
32MB	20 - 50
48MB	50 - 100
64MB	100 - 250

13.5.1.2 AS/400 CPU Requirements

The Integrated PC Server is supported by all AS/400 Advanced System and Advanced Server models. Because relatively little AS/400 system CPU is consumed on behalf of file serving requests generated by LAN Requester clients, the various AS/400 models typically yield similar file serving performance characteristics. LAN Server/400 provides more file serving throughput at significantly less CPU as compared with Client Access file serving.

To minimize the use of AS/400 CPU resource when doing read operations from Integrated PC Server-attached clients, large file system caches should be used on the Integrated PC Server. The following section on DASD I/O Requirements discusses HPFS cache sizes for the Integrated PC Server. The data in Table 28 shows the effect on AS/400 CPU and disk resources when running a fixed workload using Integrated PC Servers of three different memory sizes.

Table 28. Effects of Integrated PC Server Memory on AS/400 Resources for Fixed Workload

LAN Server/400-Integrated PC Server AS/400 400-2131 V3R6			
Integrated PC Server Configuration	**AS/400 CPU Utilization**	**AS/400 I/Os per second**	**BAPCo5 Average Completion Time**
16MB	8.0	19.5	719
32MB	4.5	10.9	691
64MB	3.3	7.5	680

Note:
Average utilization of 16 Mbps LAN was 9% for all tests.
Average utilization of Integrated PC Server ranged from 28% to 32%.

13.5.1.3 AS/400 System DASD I/O Requirements

To minimize the impact on the AS/400 disk I/Os, it is best to make maximum use of the HPFS cache on the Integrated PC Server by ensuring that sufficient Integrated PC Server memory is available for HPFS cache. The data in Table 28 on page 337 shows fewer disk I/Os happening with larger Integrated PC Server memory, since the HPFS cache is larger. HPFS cache is used by the High Performance File System for data being read from and written to disk. In *LAN Server/400 - A Guide to using the AS/400 as a File Server*, GG24-4378, a detailed description is given on how the memory on the Integrated PC Server is allocated. HPFS memory allocations for various Integrated PC Server configurations are shown in the following table:

Table 29. AS/400 Integrated PC Server HPFS Cache Sizes

Integrated PC Server Memory Size (MB)	HPFS Cache Size (MB)	
	1 Port	2 Ports
16	4.8	2.8
32	16.3	13.8
48	28.6	25.6
64	43.6	40.6

Notice that for the 16MB Integrated PC Server with two ports, less than 3MB is available for HPFS cache.

HPFS Cache statistics and other performance counters are available from the Integrated PC Server. This data can be collected using the STRPFRMON command. *OS/400 Work Management Version 3*, SC41-4306, provides information on the data collected by the STRPFRMON command.

13.5.1.4 AS/400 System Memory Requirements (Machine Pool)

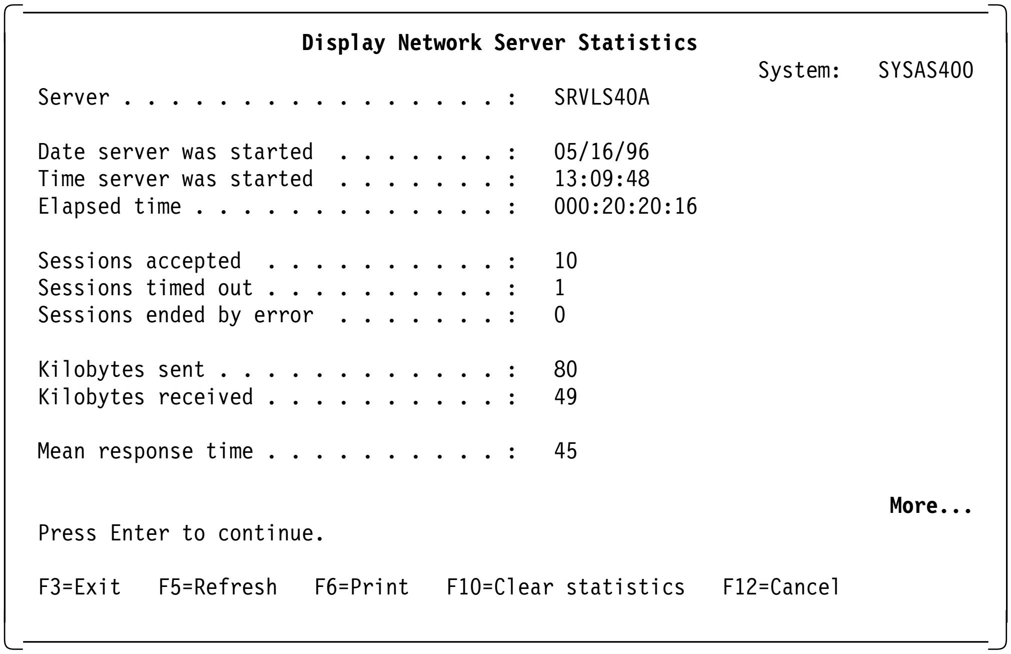
Using an Integrated PC Server with one port requires approximately 2700k bytes from the AS/400 system pool #1 (machine pool) exclusively for Integrated PC Server and LAN Server/400. Using an Integrated PC Server with two ports requires approximately 4200k bytes from system pool #1 exclusively for Integrated PC Server and LAN Server/400. These memory requirements are different from the memory on the Integrated PC Server card itself.

13.5.2 LAN Server/400 Tips and Analysis

This section describes how to measure some key LAN Server/400 statistics and also describes how to change parameters to optimize performance.

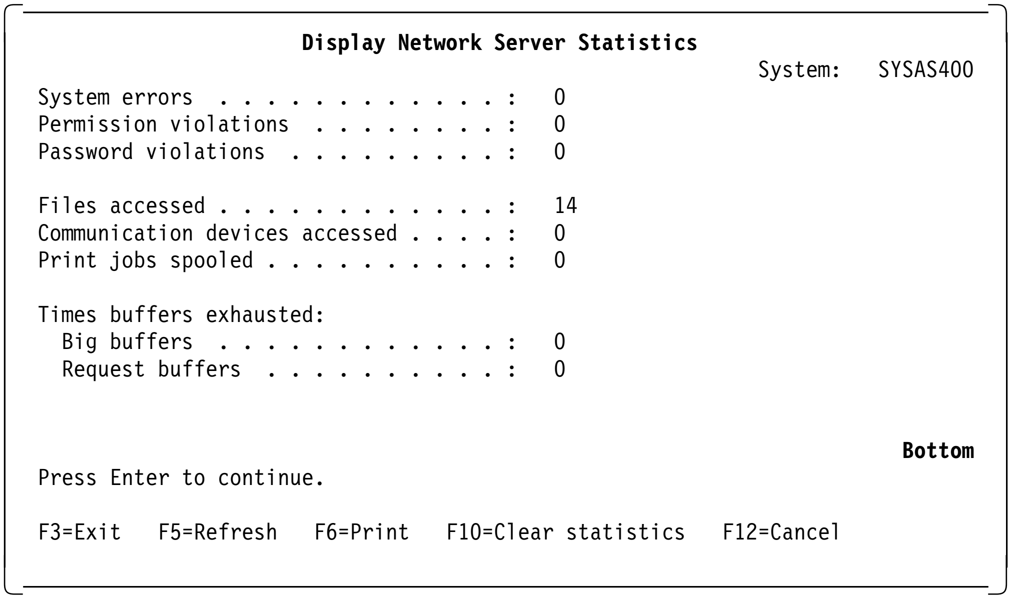
13.5.2.1 Displaying Network Statistics with LAN Server/400

Displaying network statistics and error logs can help you determine how your server is performing. This should be done during peak server workload periods. Figure 50 on page 339 and Figure 51 on page 339 display the results of the DSPNWSSTC command entered from the AS/400 command line.



```
                        Display Network Server Statistics
                                                            System:   SYSAS400
Server . . . . . . . . . . . . . . . . :   SRVLS40A

Date server was started  . . . . . . . :   05/16/96
Time server was started  . . . . . . . :   13:09:48
Elapsed time . . . . . . . . . . . . . :   000:20:20:16

Sessions accepted  . . . . . . . . . . :   10
Sessions timed out . . . . . . . . . . :   1
Sessions ended by error  . . . . . . . :   0

Kilobytes sent . . . . . . . . . . . . :   80
Kilobytes received . . . . . . . . . . :   49

Mean response time . . . . . . . . . . :   45

                                                                       More...
Press Enter to continue.

F3=Exit   F5=Refresh   F6=Print   F10=Clear statistics   F12=Cancel
```

Figure 50. Display Network Server Statistics - Display 1



```
                        Display Network Server Statistics
                                                            System:   SYSAS400
System errors  . . . . . . . . . . . . :   0
Permission violations  . . . . . . . . :   0
Password violations  . . . . . . . . . :   0

Files accessed . . . . . . . . . . . . :   14
Communication devices accessed . . . . :   0
Print jobs spooled . . . . . . . . . . :   0

Times buffers exhausted:
  Big buffers  . . . . . . . . . . . . :   0
  Request buffers  . . . . . . . . . . :   0


                                                                        Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F6=Print   F10=Clear statistics   F12=Cancel
```

Figure 51. Display Network Server Statistics - Display 2

Here is a brief description of the fields in Figure 50 and Figure 51:

Sessions accepted

The total number of sessions started since the *FROM* date and time.

Sessions timed-out

The total number of sessions timed out since the *FROM* date and time.

Sessions ended by error
The total number of sessions terminated by an error since the *FROM* date and time.

Kilobytes sent
The total number of kilobytes sent from the server since the *FROM* date and time.

Kilobytes received
The total number of kilobytes received from the server since the *FROM* date and time.

Mean response time
The average response time in milliseconds of the last request processed by the server. This is the time processed by the server and not the response time of the network.

System errors
The total number of system errors that occurred since the *FROM* date and time. System errors are errors from OS/2 and start with a prefix of "SYS."

Permission violations
The number of users trying to access aliases without the correct authorization since the *FROM* date and time.

Password violations
The passwords that were incorrectly entered since the *FROM* date and time.

Files accessed
The total number of file opens from all requesters since the *FROM* date and time.

Communication device accessed
The total number of communication ports accessed from all requesters since the *FROM* date and time. LAN Server/400 does not support communication ports on the Integrated PC Server. Therefore, this value is zero for an Integrated PC Server.

Print jobs spooled
The total number of printouts spooled from all requesters since the *FROM* date and time. LAN Server/400 does not support printing on the Integrated PC Server. Therefore, this value is zero for an Integrated PC Server.

Times buffers exhausted
The number of times the server buffers became full:

- Large buffers:

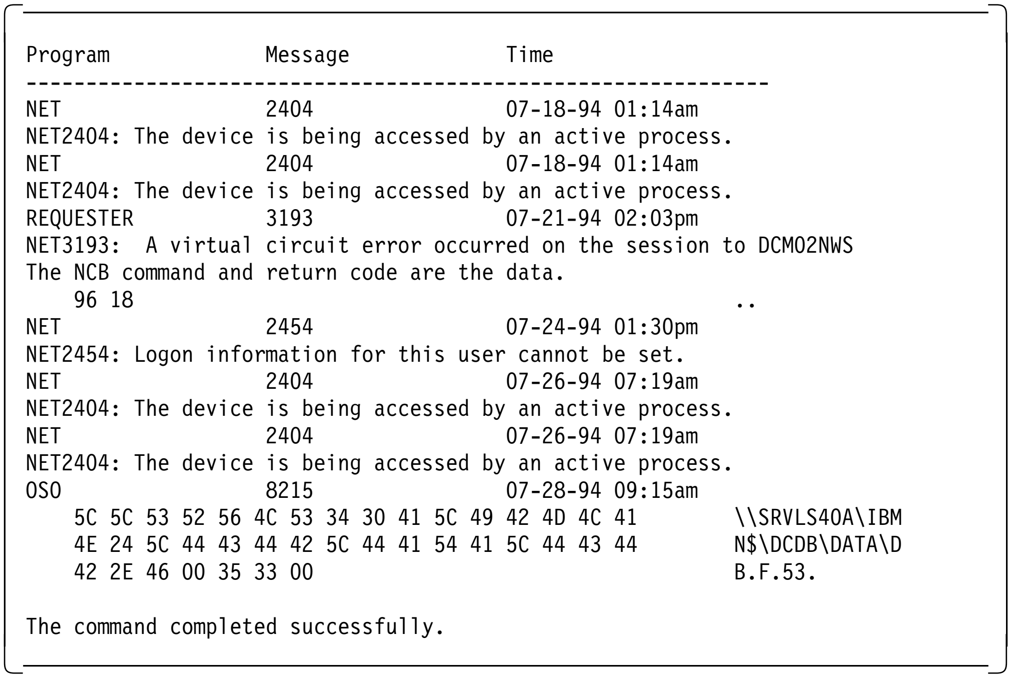
  The number of 64KB buffers the server uses for moving large files or large amounts of data. These buffers are used when loading programs or copying files from the server to the requester.
- Request buffers:

  The number of buffers the server uses to take requests from the requesters. This value should reflect the number of buffers needed to handle a peak requester work load.

To display network errors, enter the following command on an AS/400 command line:

```
SBMNWSCMD CMD('NET ERROR') SERVER(SRVLS40A)
```

If there are any LAN Server/400 network errors, they are displayed here. Figure 52 is an example of the display network error command.



```
Program             Message             Time
---------------------------------------------------------------
NET                 2404                07-18-94 01:14am
NET2404: The device is being accessed by an active process.
NET                 2404                07-18-94 01:14am
NET2404: The device is being accessed by an active process.
REQUESTER           3193                07-21-94 02:03pm
NET3193:  A virtual circuit error occurred on the session to DCMO2NWS
The NCB command and return code are the data.
    96 18                                                  ..
NET                 2454                07-24-94 01:30pm
NET2454: Logon information for this user cannot be set.
NET                 2404                07-26-94 07:19am
NET2404: The device is being accessed by an active process.
NET                 2404                07-26-94 07:19am
NET2404: The device is being accessed by an active process.
OSO                 8215                07-28-94 09:15am
    5C 5C 53 52 56 4C 53 34 30 41 5C 49 42 4D 4C 41        \\SRVLS40A\IBM
    4E 24 5C 44 43 44 42 5C 44 41 54 41 5C 44 43 44        N$\DCDB\DATA\D
    42 2E 46 00 35 33 00                                   B.F.53.

The command completed successfully.
```

Figure 52. Display Network Errors

For more details on error messages that begin with NETxxxx, enter HELP NETxxxx on an OS/2 command line or look in the *OS/2 LAN Server Problem Determination Reference Volume 3: LAN Error Messages*, S96F-8433-00.

13.5.2.2 LAN Server/400 Performance Parameters

LAN Server/400 is a modified version of OS/2 LAN that has been optimized for the AS/400 system. OS/2 LAN allows you to tune your file server using the following files:

- CONFIG.SYS
- IBMLAN.INI
- PROTOCOL.INI

LAN Server/400, however, was specifically designed so that these files are difficult to alter. Do not change them, because every time the network server is varied on, these files are refreshed. Any changes you might have made are overwritten.

There are two areas where performance can be tuned on the AS/400 system, namely:

- Add Server Storage Link (ADDNWSSTGL) discussed in Section 13.5.2.3, “Add Server Storage Link - Tuning Parameters” on page 342
- Change NetBIOS Description (CHGNTBD) discussed in Section 13.5.2.5, “NetBIOS Description” on page 344

Token-ring and Ethernet adapter descriptions may also indirectly affect the performance or response time experienced by the users.

The OS/2 LAN Requester and DOS LAN Services requesters can be optimized for better performance. See the *Network Administrator Reference Volume 2: Performance Tuning* book for more information.

When you plan to make changes on the workstations, save and print the AS/400 LAN Server files first:

- C:CONFIG.SYS
- E:IBMLANIBMLAN.INI
- E:IBMCOMPROTOCOL.INI

The following files are used for tuning requesters. Some parameters within these files must match their related parameters on the server (LAN Server/400). Because many of the LAN Server/400 parameters are fixed when you create the server, you should match the parameters on your requester to their related parameters on LAN Server/400 (rather than the other way around).

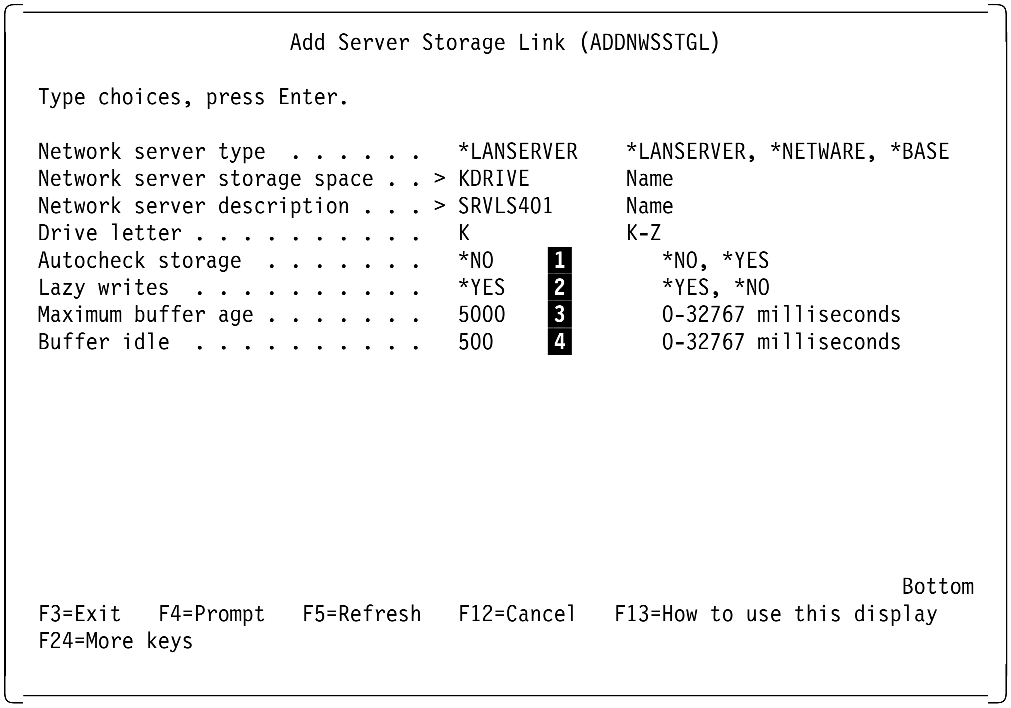
Note: The default parameters work under most circumstances. Leave your requester parameters at the default unless you have specific problems.

- CONFIG.SYS (OS/2 and DOS)
- OS/2 - IBMLAN.INI (&olr. Requester)
- OS/2. - PROTOCOL.INI (MPTS)
- DOS - PROTMAN.DOS (DOS LAN Services)
- DOS - PROTOCOL.INI (LAN Support Program - LSP)

Refer to *OS/2 Network Administrator Reference Volume 2: Performance Tuning*, S10H-9681, for a detailed explanation of the parameters affecting requester performance.

13.5.2.3 Add Server Storage Link - Tuning Parameters

There are four tuning parameters that can be entered under the ADDNWSSTGL (Add Server Storage Link) command. Enter this command and press F10 for additional parameters. Figure 53 on page 343 is an example of the result of the ADDNWSSTGL command showing the additional parameters.



```
                    Add Server Storage Link (ADDNWSSTGL)

Type choices, press Enter.

Network server type  . . . . . .   *LANSERVER    *LANSERVER, *NETWARE, *BASE
Network server storage space . . > KDRIVE        Name
Network server description . . . > SRVLS401      Name
Drive letter . . . . . . . . . .   K             K-Z
Autocheck storage  . . . . . . .   *NO      1       *NO, *YES
Lazy writes  . . . . . . . . . .   *YES     2       *YES, *NO
Maximum buffer age . . . . . . .   5000     3       0-32767 milliseconds
Buffer idle  . . . . . . . . . .   500      4       0-32767 milliseconds








                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 53. Add Server Storage Link - Tuning Parameters

1 **Autocheck Storage (CHKSTG):** LAN Server/400 always uses this value, no matter what you specify.

2 **Lazy Writes (LZYWRT):** This specifies whether this network server performs lazy writes. Using lazy writes enhances system performance by allowing cache blocks to be updated multiple times before they are written to disk. You can specify the maximum and the minimum time that data can be kept in memory before being updated to disk. Use the MAXAGE and the BUFIDLE parameters to specify these times. Lazy writes can be disabled if data integrity in the event of system failure is a major concern. Enter *YES to perform lazy writes on this server. Cache blocks are written to disk within the specified maximum age and buffer idle time values. Entering *NO forces the cached blocks to be written to disk immediately after being updated.

Lazy Write versus Caching

Do not confuse lazy writes with 386 HPFS caching. Data being handled by the lazy write function is not available to be read from memory (lazy writes are simply a technique to optimize response to the user when writing to disk).

```
Default value    *YES
Range            *NO or *YES
```

3 **Maximum Buffer Age (MAXAGE):** This specifies the maximum cache buffer age in milliseconds for this storage space. This is the maximum time data remains in the server cache before being written to disk when a lazy write cache policy is used. Keep this value larger than BUFIDLE and be careful not to make this value too low (below the default) as it defeats the objective of enabling lazy write.

If your AS/400 system has heavy disk I/O from AS/400 applications and users using LAN Server/400 require heavy disk usage, increase this value, bearing in mind that should you lose your system, there is a possibility that some data is **not** written to disk.

```
Default value    5000 milliseconds
Range            1-32267 milliseconds
```

4 **Buffer idle (BUFIDLE):** This is the cache buffer idle time in milliseconds for this storage space. This is the minimum time that data remains in the cache before being written to disk when a lazy write cache policy is used.

```
Default value    500 milliseconds
Range            1-32767 milliseconds
```

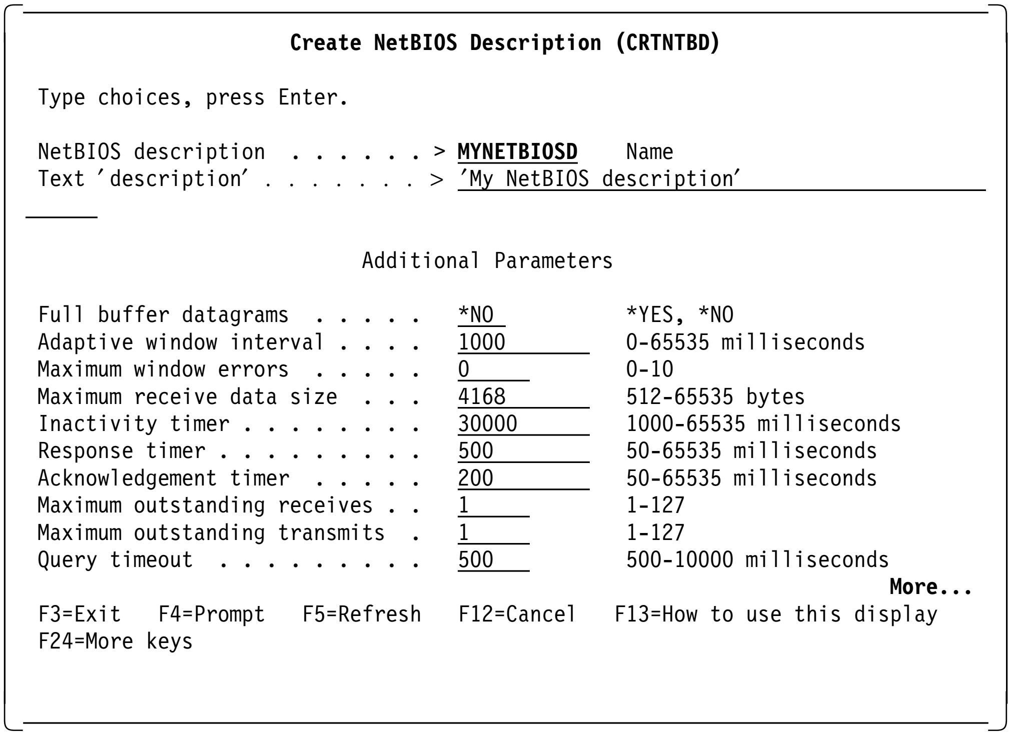
13.5.2.4 Integrated PC Server and NetBIOS

The NetBIOS support implemented in LAN Server/400 and OS/2 LAN can support up to 254 sessions per workstation. This is because the NetBIOS NCB (network control block) session number and the session number field in the NetBIOS protocol frame is a one-byte field. This NetBIOS limitation is independent of the architecture (token-ring or Ethernet) running beneath.

However, some of these sessions are required for internal LAN Server/400 use and are, therefore, not available to users. LAN Server/400 with the single-port Integrated PC Server supports up to 232 sessions on the LAN, while the dual-port version can support up to 464 simultaneous sessions.

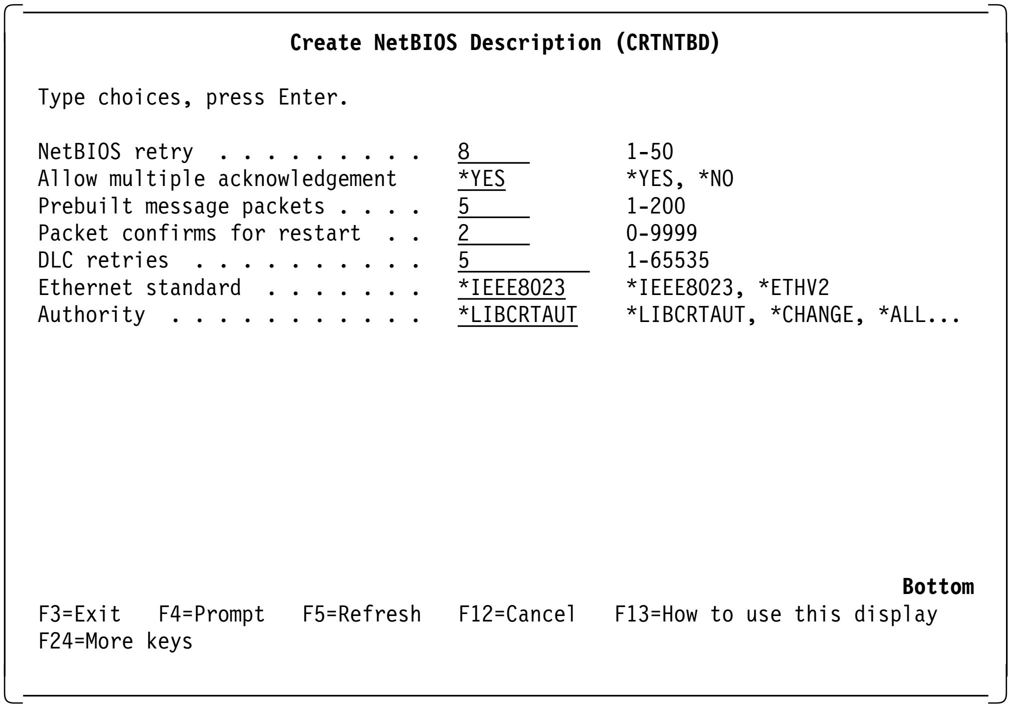
13.5.2.5 NetBIOS Description

Use the default NetBIOS description that is supplied with LAN Server/400 called QNTBIBM. Do not change these unless you are familiar with NetBIOS. If you need to create your own NetBIOS description, enter CRTNTBD from the AS/400 command line. Press F10 for additional parameters. The resulting displays are shown in Figure 54 on page 345 and Figure 55 on page 345.



```
                   Create NetBIOS Description (CRTNTBD)

 Type choices, press Enter.

 NetBIOS description  . . . . . . > MYNETBIOSD    Name
 Text 'description' . . . . . . . > 'My NetBIOS description'


                           Additional Parameters

 Full buffer datagrams  . . . . .   *NO           *YES, *NO
 Adaptive window interval . . . .   1000          0-65535 milliseconds
 Maximum window errors  . . . . .   0             0-10
 Maximum receive data size  . . .   4168          512-65535 bytes
 Inactivity timer . . . . . . . .   30000         1000-65535 milliseconds
 Response timer . . . . . . . . .   500           50-65535 milliseconds
 Acknowledgement timer  . . . . .   200           50-65535 milliseconds
 Maximum outstanding receives . .   1             1-127
 Maximum outstanding transmits  .   1             1-127
 Query timeout  . . . . . . . . .   500           500-10000 milliseconds
                                                                     More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 54. Create NetBIOS Description - 1 of 2



```
                   Create NetBIOS Description (CRTNTBD)

 Type choices, press Enter.

 NetBIOS retry  . . . . . . . . .   8             1-50
 Allow multiple acknowledgement     *YES          *YES, *NO
 Prebuilt message packets . . . .   5             1-200
 Packet confirms for restart  . .   2             0-9999
 DLC retries  . . . . . . . . . .   5             1-65535
 Ethernet standard  . . . . . . .   *IEEE8023     *IEEE8023, *ETHV2
 Authority  . . . . . . . . . . .   *LIBCRTAUT    *LIBCRTAUT, *CHANGE, *ALL...








                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 55. Create NetBIOS Description - 2 of 2

13.5.3 Impact of the Integrated PC Server on AS/400 Performance

This topic describes how the installation and operation of the Integrated PC Server affects the performance of the AS/400 system as a whole.

13.5.3.1 AS/400 Resources

AS/400 Processor: When operating with adequate memory for 386 HPFS cache, little AS/400 system CPU is used for file serving requests from Integrated PC Server-attached clients. Therefore, the various AS/400 models yield similar file serving performance characteristics regardless of their power. This is assuming, of course, that no other bottlenecks exist.

Based on preliminary measurements, an environment representing possibly 200 clients running the BAPCo5 workload experienced a CPU utilization of approximately 8% on an AS/400 model F20. This varies based on the environment and cache hit ratios.

AS/400 Machine Pool Memory: You need 2700k bytes from the AS/400 system pool (pool #1) for a one-port Integrated PC Server, and 4200k bytes for a two-port Integrated PC Server. These memory requirements are different from the memory on the Integrated PC Server card itself.

Performance Data for Integrated PC Server and LAN Server/400: The AS/400 performance monitor can be used to collect information about the performance characteristics of the Integrated PC Server and LAN Server/400. Typical job and task information is collected for Integrated PC Server-related jobs and tasks as for all other AS/400 jobs and tasks. In addition, more detailed performance information is collected by the Integrated PC Server and returned to the AS/400 system through the STRPFRMON command. Integrated PC Server performance data collected by the STRPFRMON command can be queried in the file QAPMIOPD. The performance monitor file QAPMIOPD contains a large number of statistics associated with the Integrated PC Server, including the software operating within it (LAN Server/400 and the High Performance File System, OS/2 utilizations, and Integrated PC Server CPU utilizations). For more information on QAPMIOPD file fields and descriptions, please refer to Appendix A of the *OS/400 Work Management Version 3*, SC41-4306. The information in this file is not interpreted by any of the tools into a useable report. You need to understand the implications of each of the measurements before you can use the data in this file effectively.

The AS/400 Performance Tools Component Report can be used to analyze IOP utilizations for the AS/400 system. The utilization shown in this report for an Integrated PC Server is the higher of the two utilization numbers comparing the Intel i960 processor and the Intel 486 processor. Generally, with any amount of file serving activity on the Integrated PC Server, the Intel 486 processor utilization is the one shown in the performance tools component report.

When using BEST/1, the SLIC tasks are grouped in the QDEFAULT workload unless you explicitly assign them using a user-defined workload. They can also be clearly seen on the component report provided by the Performance Tools if the Integrated PC Server is active.

BEST/1 groups these tasks in the Client Access/400 workload if there are interactive sessions coming through the LAN Server IOP and defaults are taken. The IOPs appear in the BEST/1 communications configurations and as communications IOPs. BEST/1 recognizes LAN Server/400 job types when creating models. You can specify Integrated PC Server-related tasks using the *FSIOP job type. Refer to *BEST/1 Capacity Planning Tool*, SC41-3341, for more information.

13.5.4 LAN Server/400 Save/Restore Performance

This section discusses options for backup and recovery and provides examples to help you develop and carry out your save strategy.

A major benefit you get with the Integrated PC Server and LAN Server/400 is the ability to include your LAN Server/400 backup procedure into your AS/400 backup procedure.

To save or restore the /QLANSrv file system data, you use the save and restore commands that support the integrated file system on the AS/400 system. The commands are SAV (Save) and RST (Restore). These commands use the object naming convention from the Integrated File System. For more information about the SAV command, see *OS/400 Backup & Recovery - Basic V3R6*, SC41-4304.

The Integrated File System design allows you to save files from:

- Other Integrated PC Servers
- OS/2 LAN Server systems

The files can reside on the local Integrated PC Server or on remote servers. This capability makes the AS/400 system a powerful part of the domain. You can use an AS/400 system to save the server data from any of the LAN Server systems in the domain.

To save and restore other objects that are associated with LAN Server/400, use SAVOBJ (Save Object) and RSTOBJ (Restore Object).

13.5.5 Save/Restore—Planning

To begin developing a strategy, you must decide what to save and how often to save. Before you decide on a strategy, you must be know about several LAN Server/400 objects and their contents.

13.5.5.1 LAN Server/400 Structure

You can find pieces of LAN Server/400 in several parts of the AS/400 system:

Library QXZ1: QXZ1 holds a number of objects; three of these objects are the base from which the AS/400 system creates network server descriptions.

Library QUSRSYS: The disk images for each network server description are stored here. There are two storage spaces for each network server description. For the network server description in the example, ″SRVLS40A″, they are called SRVLS40A1.SVRSTG and SRVLS40A3.SVRSTG. The names consist of the server name followed by a one or a three. SRVLS40A1 holds the files and programs to boot the Integrated PC Server; SRVLS40A3 holds the domain control database information.

Library QSYS29nn (Where 29nn is a Language Number): This library contains licensed system code for secondary languages. This library holds the national language versions of the code that is stored in QXZ1. It contains two objects, QFPHSYS2.SVRSTG and QFPHSYS3.SVRSTG (note that QFPHSYS1.SVRSTG does not have an NLS version).

Integrated File System Directory /QFPNWSSTG: /QFPNWSSTG holds the storage spaces that you link to the network server descriptions.

In this view, the storage spaces are seen as solid blocks of data (there is no way to see individual files or directories).

Integrated File System Directory /QLANSrv: This directory contains the LAN Server/400 file system as a hierarchical structure of directories and files that can be saved and restored individually or in groups.

Integrated File System Directory /QLANSrvSR (V3R1): This directory is a temporary storage area for files that are in the process of being saved to AS/400 tapes. The QLANSrvSR directory exists on an AS/400 system running V3R6 or V3R2 if you upgraded your AS/400 system from V3R1. OS/400 does not use the directory in either V3R6 or V3R2.

13.5.5.2 Defining Your Save/Restore Strategy for QLANSrv Data

When defining save/restore procedures for the network drives in the QLANSrv file system, there are two options:

1. Saving/restoring at the network drive level
2. Saving/restoring at the file/directory level on a network drive

From a performance perspective, option 1 is dramatically faster than option 2. For option 1, the save/restore data rate was measured at about 2GB to 8GB per hour. For option 2, the save/restore data rate was approximately 75MB to 500MB per hour. (The actual performance achieved is dependent on a number of factors. Please refer to the following sections for more detailed performance information.) These data rates were achieved with the QLANSrv save/restore PTFs, which improve save performance by about 20% and restore performance by about 15%.

However, there are some considerations to be aware of. Saving at the network drive level is similar to a Save Storage (SAVSTG) operation. Therefore, if a restore is required, the entire network drive must be restored. Also, in order to use option 1, the Integrated PC Server that the network drive is linked to must be varied off to perform the save.

In regards to option 2, if you save the network drive at the file/directory level, and if a restore is required, you can selectively restore only the files you need. Also, the Integrated PC Server does not need to be varied off when saving/restoring at the file/directory level. Save/restore of a network drive at the file/directory level is about two and one-half to three times (2.5-3X) slower than saving/restoring data (folders/documents) in the QDLS file system using the SAVDLO/RSTDLO commands.

Therefore, taking into account the preceding considerations, we recommend the following for saving/restoring network drives in QLANSrv:

- When configuring the network drives for an Integrated PC Server, partition your volatile data (changing data) and static data (such as programs or archived data) on separate network drives.
- If possible, use option 1 to save network drives that have static data on them. This is probably done at less frequent intervals than saving volatile data.

- For network drives with volatile data, if you have the requirement to restore at the file/directory level, or you need 100% availability of the Integrated PC Server, you must use option 2. However, if option 2 does not provide the required performance, you should consider the following:
  - Save using option 1 (save at the network drive level). You must vary off the Integrated PC Server during the save.
  - When a restore is required, create a temporary network drive in QLANSrv and restore into the temporary network drive.
  - Selectively restore required files from the temporary network drive.
  - Delete the temporary drive when complete.

13.5.6 Save/Restore Strategies

Because LAN Server/400 uses various pieces of the AS/400 system, your strategy for saving and restoring LAN Server/400 and the data it manages is going to depend on your company′s needs. As mentioned in Section 13.5.4, "LAN Server/400 Save/Restore Performance" on page 347, one of the major benefits of LAN Server/400 over PC file servers is the ability to include your LAN Server/400 backup procedure into the AS/400 backup procedure using the SAVSYS command. You can do this as often as you currently save the entire AS/400 system. ***/QFPNWSSTG*** and ***/QLANSrv*** offer you two methods to save the entire AS/400 system. The second method performs better because the following SAV command example must verify internally that AS/400 libraries and other objects are to be saved.

You can also save your LAN Server/400 data by saving the entire storage space, or you can save a portion of the storage space, such as a directory or a file.

13.5.6.1 Performance Impact

Your save/restore strategy can have significant impacts on the time it takes to save the AS/400 system. Consider this carefully.

Note: Saving the entire storage space through /QFPNWSSTG is **significantly faster** than saving only a portion of the storage space through /QLANSrv unless you are saving only a small portion of the storage space. For example, when saving through /QFPNWSSTG, the save/restore data rate was measured at about 2GB to 8GB per hour[1]. When saving through /QLANSrv the save/restore data rate was approximately 75MB to 500MB per hour[1].

However, if you must later restore the data, you must restore the entire storage space. If you save only a portion of the storage space, you can restore only the files you need. Also, you do not have to vary off the Integrated PC Server to perform this type of save operation.

Finally, also note that if you save or restore a directory from a storage space, performance is two and one-half to three times slower than saving a folder and its documents using the SAVDLO command and RSTDLO command.

[1] The actual data rate achieved depends on a number of factors. Things that affect results are CPU model, the tape drive you are using, the pool size, and the size of the files you are saving.

13.5.6.2 Save Regularly

Your backup strategy should include saving the storage spaces (also sometimes called network drives) regularly. Create several storage spaces. Store data that change infrequently on different storage spaces from data that change frequently. Using this strategy, you can save the storage space containing the infrequently changed applications or data less often, perhaps only when you save the entire AS/400 system. The following sections provide examples and tips on how to save the entire system, storage spaces only, and the other objects that are part of the LAN Server/400 product.

It is important that you plan the frequency of your saves to ensure that you always have a usable backup available in the event of a system failure or disaster. For example, you may decide on the following strategy to ensure that your user data is thoroughly backed up.

13.5.6.3 Memory Tuning for Save/Restore Performance

The /QLANSrv is sensitive to the size of the AS/400 memory pool. To achieve acceptable save performance, do the following calculations. Each of the two calculations must have a result greater than or equal to 15MB:

- The pool size divided by 2
- The pool size divided by the activity level of the pool

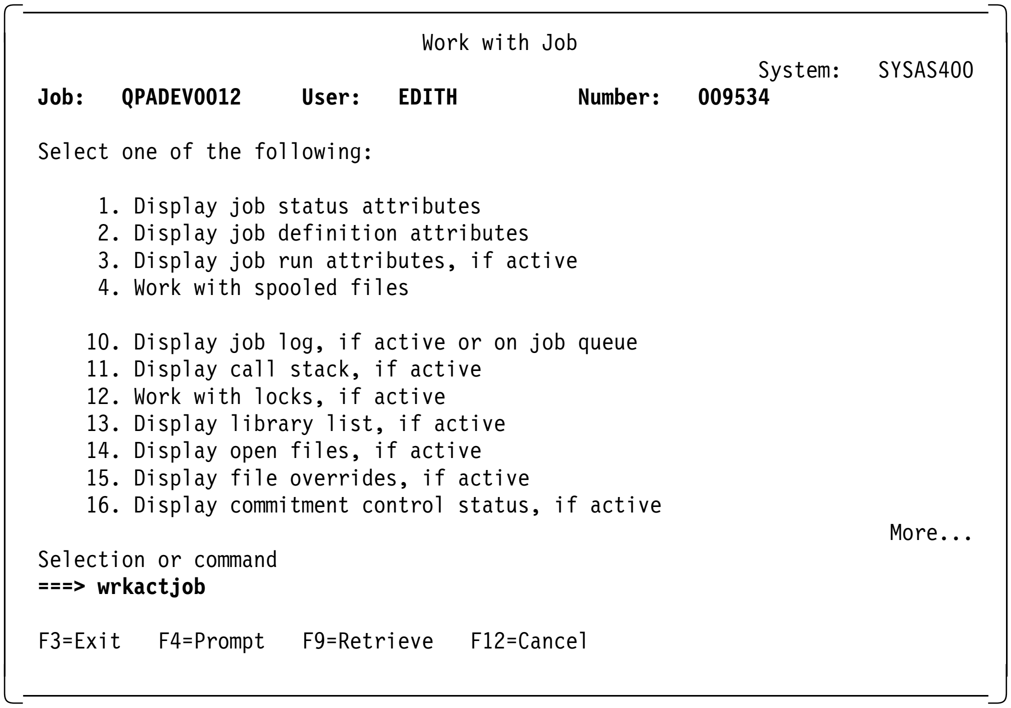
If insufficient memory is allocated to the pool where save or restore is running, the process takes much longer to complete.

To calculate memory for saving the system:

1. Determine which pool the job that must run the save or restore operation runs in.

   For example, user EDITH plans to run the SAV command interactively. Using the WRKJOB command, you can determine the job ID, user ID, and job number for EDITH. They are QPADEV0012, EDITH, and 009534 (Figure 56 on page 351).

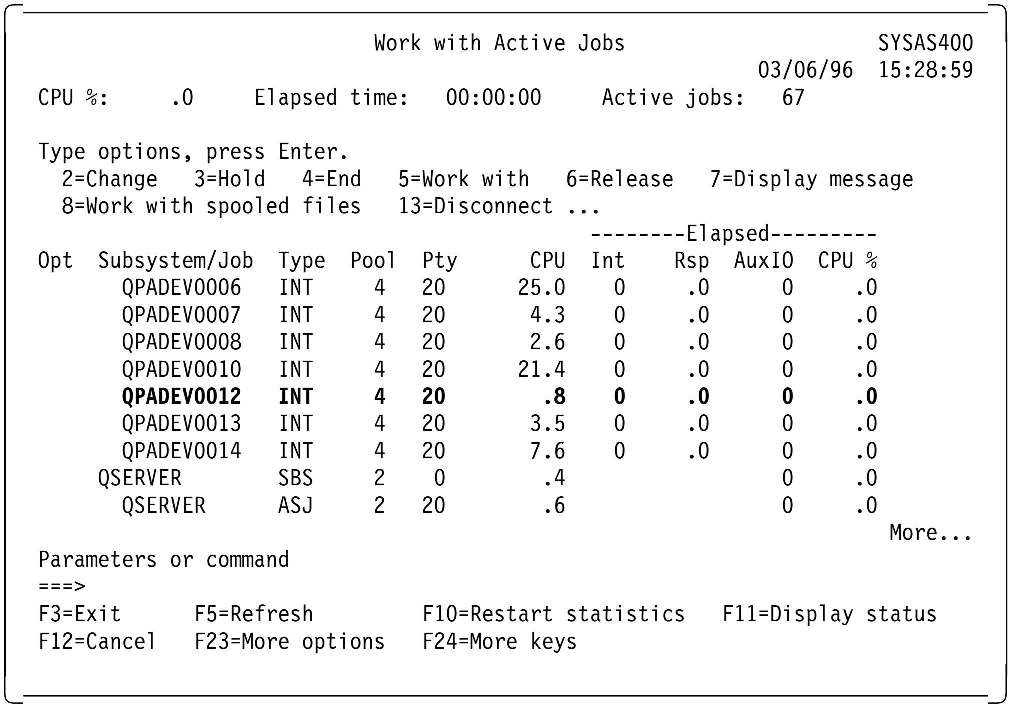
   Now, find the pool in which the job runs by entering the WRKACTJOB command.



```
                                Work with Job
                                                            System:   SYSAS400
 Job:   QPADEV0012     User:   EDITH          Number:   009534

 Select one of the following:

      1. Display job status attributes
      2. Display job definition attributes
      3. Display job run attributes, if active
      4. Work with spooled files

     10. Display job log, if active or on job queue
     11. Display call stack, if active
     12. Work with locks, if active
     13. Display library list, if active
     14. Display open files, if active
     15. Display file overrides, if active
     16. Display commitment control status, if active
                                                                       More...
 Selection or command
 ===> wrkactjob

 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 56. Work with Job Display — Finding Job Information

Press F11 on the Work with Active Jobs display until the display that is shown in Figure 57 appears. Notice that job EDITH (QPADEV0012) is using pool 4.



```
                            Work with Active Jobs                     SYSAS400
                                                            03/06/96  15:28:59
 CPU %:     .0     Elapsed time:   00:00:00     Active jobs:   67

 Type options, press Enter.
   2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
   8=Work with spooled files   13=Disconnect ...
                                                     --------Elapsed---------
 Opt  Subsystem/Job  Type  Pool  Pty      CPU  Int    Rsp  AuxIO  CPU %
        QPADEV0006   INT     4   20      25.0    0     .0      0     .0
        QPADEV0007   INT     4   20       4.3    0     .0      0     .0
        QPADEV0008   INT     4   20       2.6    0     .0      0     .0
        QPADEV0010   INT     4   20      21.4    0     .0      0     .0
        QPADEV0012   INT     4   20        .8    0     .0      0     .0
        QPADEV0013   INT     4   20       3.5    0     .0      0     .0
        QPADEV0014   INT     4   20       7.6    0     .0      0     .0
      QSERVER        SBS     2    0        .4                  0     .0
        QSERVER      ASJ     2   20        .6                  0     .0
                                                                       More...
 Parameters or command
 ===>
 F3=Exit      F5=Refresh        F10=Restart statistics   F11=Display status
 F12=Cancel   F23=More options   F24=More keys
```

Figure 57. Work with Active Jobs — Finding the System Pool

Now we know which pool to use for the calculations.

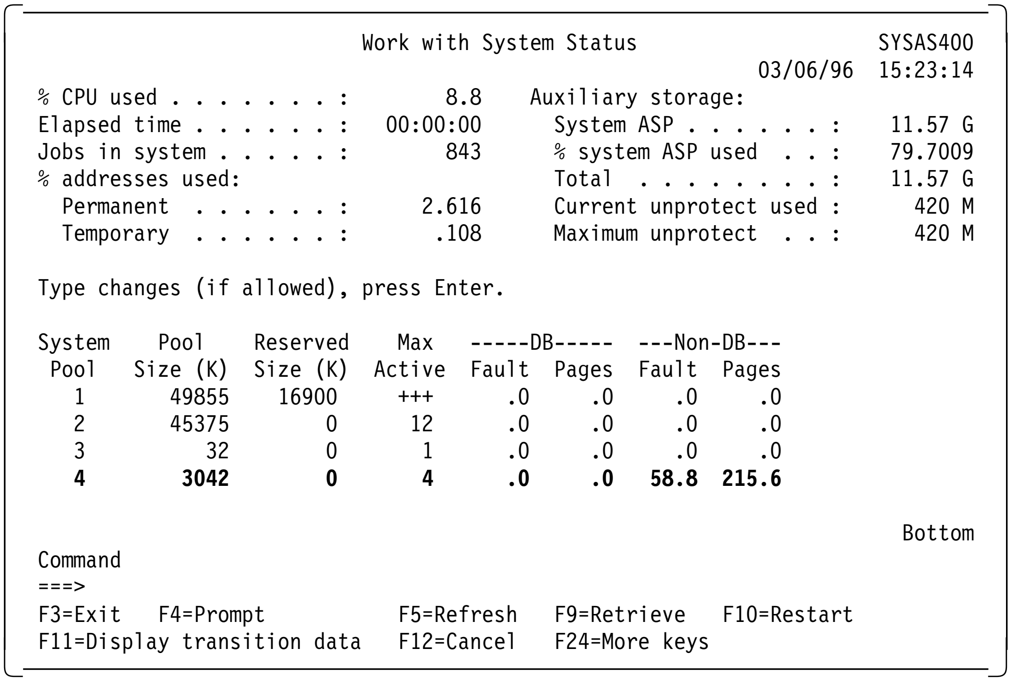
2. Calculate whether the pool has enough memory to run the job efficiently.

Use the WRKSYSSTS command to display the system storage pools (shown in Figure 58 on page 352). As previously discussed, use the following two calculations to determine if adequate memory is allowed for the save operation. For each calculation, the storage pool size for the storage pool your job uses must be at least 15MB.

- The pool size divided by 2
- The pool size divided by the activity level of the pool

For example, if the pool size is 60MB and the activity level is 6, the preceding two calculations are 10MB (pool size divided by the activity level), and 30MB (pool size divide 2). In this case, either increase the pool size to 90MB or decrease the activity level to 4. In Figure 58, system pool 4 has only 3043KB (3MB) of storage.

There is no need to perform the second calculation at this time. 3042K is not enough to begin with. Therefore, dividing that number by 4 emphasizes that a SAV job running in this amount of memory is going to take a **long** time.



```
                           Work with System Status                   SYSAS400
                                                             03/06/96  15:23:14
% CPU used . . . . . . . :        8.8    Auxiliary storage:
Elapsed time . . . . . . :   00:00:00      System ASP . . . . . . :    11.57 G
Jobs in system . . . . . :        843      % system ASP used  . . :    79.7009
% addresses used:                          Total  . . . . . . . . :    11.57 G
  Permanent  . . . . . . :      2.616      Current unprotect used :      420 M
  Temporary  . . . . . . :       .108      Maximum unprotect  . . :      420 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   -----DB-----  ---Non-DB---
 Pool   Size (K)  Size (K)  Active  Fault  Pages  Fault  Pages
   1      49855     16900     +++      .0     .0     .0     .0
   2      45375         0      12      .0     .0     .0     .0
   3         32         0       1      .0     .0     .0     .0
   4       3042         0       4      .0     .0   58.8  215.6

                                                                        Bottom
Command
===>
F3=Exit   F4=Prompt           F5=Refresh   F9=Retrieve   F10=Restart
F11=Display transition data   F12=Cancel   F24=More keys
```

Figure 58. Work with System Status — Displaying System Pool Memory

Notice that storage pool 1 and storage pool 2 have plenty of storage; therefore, you can now do one of the following:

- Increase the pool size.
- Decrease the activity value.
- Do a combination of the two.

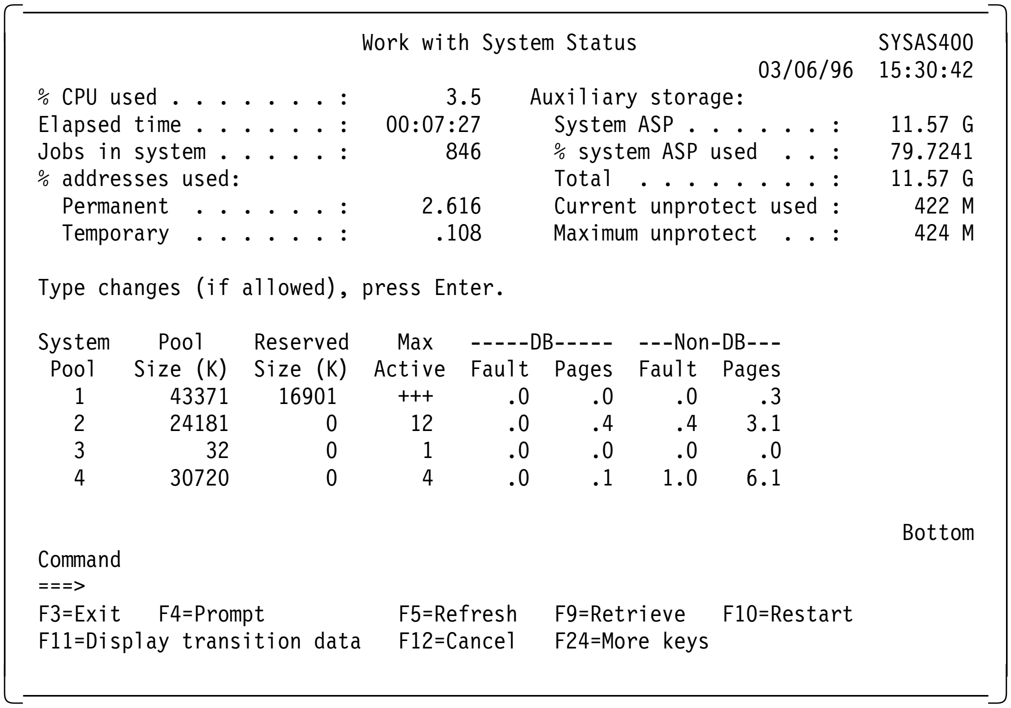
In Figure 59 on page 353, the pool size was changed to 30720KB. Is this enough? The first calculation yields enough memory:

```
30720 divided by 2 yields 15MB.
```

Now, the second calculation; the pool size divided by the activity level:

```
30720 divided by 4 yields 7500  (7.68MB)
```

More storage is needed. You may want to decrease the maximum activity level to 2 for this pool.



```
                             Work with System Status                  SYSAS400
                                                            03/06/96  15:30:42
 % CPU used . . . . . . . :        3.5    Auxiliary storage:
 Elapsed time . . . . . . :   00:07:27      System ASP . . . . . . :    11.57 G
 Jobs in system . . . . . :        846      % system ASP used  . . :    79.7241
 % addresses used:                          Total  . . . . . . . . :    11.57 G
   Permanent  . . . . . . :      2.616      Current unprotect used :      422 M
   Temporary  . . . . . . :       .108      Maximum unprotect  . . :      424 M

 Type changes (if allowed), press Enter.

 System    Pool    Reserved    Max   -----DB-----  ---Non-DB---
  Pool   Size (K)  Size (K)  Active  Fault  Pages  Fault  Pages
    1       43371    16901     +++      .0     .0     .0     .3
    2       24181        0      12      .0     .4     .4    3.1
    3          32        0       1      .0     .0     .0     .0
    4       30720        0       4      .0     .1    1.0    6.1

                                                                        Bottom
 Command
 ===>
 F3=Exit   F4=Prompt           F5=Refresh   F9=Retrieve   F10=Restart
 F11=Display transition data   F12=Cancel   F24=More keys
```

Figure 59. Work With System Status — Adjusting Pool Size

Experiment with these values to ensure that your save and restore processes are running as fast as they can. Ensure that you are not affecting other operations on the system by taking memory for the save operation. Other factors, such as the size of your files, tape speed, disk arms, and your processor feature also influence the speed at which the AS/400 system can save or restore your data. For more information on managing system activities, see the *OS/400 Work Management Version 3*, SC41-4306.

If less than the recommended memory is allocated to the pool where the save is running, the save operation may take significantly longer. Also, if there is other work being run in the pool you may have to increase the pool size. For example, the job EDITH was not the only one running in system pool 4 in Figure 57 on page 351. Use the guideline calculations as a starting point. For restore operations, start with the same calculations and adjust the pool size, if necessary, to achieve the acceptable restore performance.

You should refer to *OS/400 Work Management Version 3*, SC41-4306, for information on adjusting pool sizes and activity levels.

If less than the recommended memory is allocated to the pool where the save is executing, the save operation may take significantly longer. Also, if there is other work being executed in the pool, you may have to increase the pool size. Use the preceding guidelines as a starting point. For restores, you should use the save guidelines as a starting point and adjust the pool size, if necessary, to achieve the acceptable restore performance.

The data rate achieved for saving and restoring QLANSrv data is highly dependent on a number of factors such as file size, processor feature, tape speed, and DASD arms. This makes predicting save/restore times difficult, and your results may vary from the results discussed here.

13.5.7 Network Drive Level Save/Restore Data Rates

Saving/Restoring QLANSrv data at the network drive level is dramatically faster than saving at the file/directory level. For example, on a Model 510/2144 with a 3490E tape drive, the save rate at the network drive level is approximately 8150MB per hour versus 350MB per hour at the file/directory level. On a model 400/2133, the respective data rates are 7400MB per hour versus 295MB per hour with a 3490E tape drive. The tape drive and CPU are the dominant factors in this case, and slower tape drives result in slower save/restore rates.

1. The performance difference is dependent on the type of data being saved/restored and system characteristics.
2. Use the WRKSYSSTS command to determine where the pool size and activity level should be set for the save operation.

13.5.8 Integrated PC Server Performance Monitor Data Queries

Figure 60 on page 355 shows the query definition when selecting ″HPFS data″ from Performance Monitor database file QAPMIOPD. Figure 61 on page 356 shows the query output for a specific collection of Integrated PC Server ″HPFS data″ from Performance Monitor database file QAPMIOPD. Refer to *OS/400 Work Management Version 3*, SC41-4306, for field definitions. Field XIDTYP containing a ″3″ identifies HPFS data.

```
5763QU1  V3R6M0  940909           IBM Query/400            GENSYS02  3/13/96   16:53:36         Page    1
  Query . . . . . . . . . . . . . . . . . HPFSCOOK02
    Library . . . . . . . . . . . . . . . DJOHNSON
  Query text  . . . . . . . . . . . . . . HPFS386 Statistics- Cook 02
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . US
    *** . is the decimal separator character for this query ***
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . No
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes

Selected files
  ID     File           Library       Member        Record Format
  T01    QAPMIOPD       FSCITY2       CPUTEST2      QAPMIOXR
Result fields
  Name        Expression                         Column Heading          Len  Dec
  TOTALREADS  XICT01 + XICT02                    Total #                  11    0
                                                 READS
  CACHEHITRD  100 * XICT01 / (XICT01 + XICT02)   CACHE                     5    2
                                                 HIT %
                                                 READS
  TOTALWRITE  XICT03 + XICT04                    Total #                   7    0
                                                 Writes
  CACHEHITWR  100 * XICT04 / (XICT03 + XICT04)   CACHE                     7    2
                                                 HIT
                                                 WRITE
Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           XIDTYP           EQ       '3'
   AND     XICT01           GT       0
   AND     XICT04           GT       0


                                  IBM Query/400                      3/13/96   16:53:36         Page    2
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  INTNUM                                       Interval Number
  INTSEC                                       Elapsed Interval Seconds
  XIIOPA                                       IOP Bus Address
  XIDTYP                                       Type of data in record
  TOTALREADS
  CACHEHITRD
  TOTALWRITE
  CACHEHITWR
  XICT10                                       Counter 10
  XICT11                                       Counter 11
  XICT01                                       Counter 01
  XICT02                                       Counter 02
  XICT03                                       Counter 03
  XICT04                                       Counter 04
Report column formatting and summary functions
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count          Overrides
  Field           Summary    Column                              Dec  Null       Dec  Numeric
  Name            Functions  Spacing  Column Headings        Len  Pos  Cap   Len  Pos  Editing
  INTNUM                     0        Int                      5    0          1    0
                                      #
  INTSEC                     1        Int                      7    0          5    0
                                      Sec
  XIIOPA                     1        IOP                      3    0
                                      Bus
  XIDTYP                     1                                 1
                                      Data
                                      Type
  TOTALREADS      1          0        Total #                 11    0
                                      READS
  CACHEHITRD      2          0        CACHE                    5    2          5    2
                                      HIT %
                                      READS
  TOTALWRITE      1          0        Total #                  7    0
                                      Writes
  CACHEHITWR      2          0        CACHE                    7    2          7    2
                                      HIT %
                                      WRITE
```

Figure 60 (Part 1 of 2). QAPMIOPD File Query Definition - CACHE

```
  XICT10          2          0        Files                  11    0          6    0
                                      Opened
  XICT11          2          0        Files                  11    0          6    0
                                      Closed
  XICT01          1          0        Read Reqs              11    0         11    0
                                      frm CACHE


                                   IBM Query/400                      3/13/96   16:53:36         Page    3
Report column formatting and summary functions (continued)
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count           Overrides
  Field           Summary    Column                                Dec  Null       Dec  Numeric
  Name            Functions  Spacing  Column Headings        Len  Pos  Cap   Len  Pos  Editing
  XICT02          1          0        Read Reqs              11    0
                                      from DISK
  XICT03          1          0        Write reqs             11    0
                                      from DISK
  XICT04          1          0        Write Reqs             11    0
                                      LAZY
                                      Written
Selected output attributes
  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . Yes
    Wrapping width  . . . . . . . . . . . 168
    Record on one page  . . . . . . . . . No

Printer Output
  Printer device  . . . . . . . . . . . . PRT03
  Report size
    Length  . . . . . . . . . . . . . . .  66
    Width . . . . . . . . . . . . . . . . 166
  Report start line . . . . . . . . . . .   6
  Report end line . . . . . . . . . . . .  60
  Report line spacing . . . . . . . . . . Double space
  Print definition  . . . . . . . . . . . Yes

Printer Spooled Output
  Spool the output  . . . . . . . . . . . Yes
  Form type . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Copies  . . . . . . . . . . . . . . . .   1
  Hold  . . . . . . . . . . . . . . . . . Yes

Cover Page
  Print cover page  . . . . . . . . . . . Yes
    Cover page title
      8 Users LAN Server/400 - QAPMIOPD(HPFS) Read, Write Cache, File Open/Close


                                   IBM Query/400                      3/13/96   16:53:36         Page    4
Page headings and footings
  Print standard page heading . . . . . . Yes
    Page heading
      LAN Server/400: HPFS Cache Statistics and File Open/Close
    Page footing
```

Figure 60 (Part 2 of 2). QAPMIOPD File Query Definition - CACHE

```
                                             8 Users LAN Server/400 - QAPMIOPD(HPFS) Read, Write Cache, File Open/Close
                                                        QUERY NAME . . . . . HPFSCOOK02
                                                        LIBRARY NAME . . . . DJOHNSON
                                                        FILE         LIBRARY      MEMBER       FORMAT
                                                        QAPMIOPD     FSCITY2      CPUTEST2     QAPMIOXR
                                                        DATE . . . . . . . . 03/13/96
                                                        TIME . . . . . . . . 16:53:37
                                                                   HPFS386 Statistics- Cook 02


03/13/96  16:53:37                                    LAN Server/400: HPFS Cache Statistics and File Open/Close      PAGE    1
Int    Int  IOP                 Total #  CACHE       Total #   CACHE  Files    Files    Read Reqs  Read Reqs  Write reqs  Write Reqs
#      Sec  Bus  Data           READS    HIT %       Writes    HIT %  Opened   Closed   frm CACHE  from DISK  from DISK   LAZY
                 Type                    READS                 WRITE                                                    Written
 1     301    2   3              95,347  87.41         6,933  100.00     995      975      83,351     11,996           0      6,933
 2     298    2   3             232,309  95.12         3,136  100.00   3,969    3,974     220,981     11,328           0      3,136
 3     298    2   3             222,045  93.72         7,537   99.98   3,980    3,995     208,117     13,928           1      7,536
                FINAL TOTALS
                TOTAL           549,701               17,606                              512,449     37,252           1     17,605
                AVG                      92.08                 99.99   2,981    2,981
* * *  E N D  O F  R E P O R T  * * *
```

Figure 61. QAPMIOPD File Query Report - CACHE

Figure 62 on page 357 shows the query definition when selecting ″CPU data″ from Performance Monitor database file QAPMIOPD. Figure 63 on page 358 shows the query output for a specific collection of Integrated PC Server ″CPU data″ from Performance Monitor database file QAPMIOPD. Refer to *OS/400 Work Management Version 3*, SC41-4306, for field definitions. Field XIDTYP containing a ″2″ identifies Integrated PC Server CPU data.

```
5763QU1  V3R6M0  940909           IBM Query/400            GENSYS02  3/22/96   13:59:07        Page    1
  Query . . . . . . . . . . . . . . . . . CPU486COOK
    Library . . . . . . . . . . . . . . . DJOHNSON
  Query text  . . . . . . . . . . . . . . Integrated PC Server 486 CPU Utilizationt Statistics
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . US
    *** . is the decimal separator character for this query ***
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . No
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes

Selected files
  ID     File           Library       Member        Record Format
  T01    QAPMIOPD       FSCITY2       BAP8WS64MB    QAPMIOXR
Result fields
  Name        Expression                       Column Heading          Len  Dec
  CPU486INT   ((XICT01 / 1000) / INTSEC) * 100   486 CPU %                 7    1
                                                 Utilization
  XICT01SECS  XICT01 / 1000                      486 CPU                   7    1
                                                 Seconds
Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or ′Characters′)
           XIDTYP           EQ       ′2′
   AND     XICT01           NE       0
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  INTNUM                                       Interval Number


                                  IBM Query/400                      3/22/96   13:59:07        Page    2
Ordering of selected fields (continued)
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  XIIOPA                                       IOP Bus Address
  XIDTYP                                       Type of data in record
  XITYPE                                       IOP Type
  INTSEC                                       Elapsed Interval Seconds
  XICT01SECS
  CPU486INT
Report column formatting and summary functions
  Summary functions:  1-Total, 2-Average, 3-Minimum, 4-Maximum, 5-Count          Overrides
  Field           Summary    Column                                Dec  Null       Dec  Numeric
  Name            Functions  Spacing  Column Headings         Len  Pos  Cap   Len  Pos  Editing
  INTNUM                     0        Int                       5    0          1    0
                                      #
  XIIOPA                     1        IOP                       3    0
                                      Bus
  XIDTYP                     1        Data                      1
                                      Type
                                      (OS/2)
  XITYPE                     2        IOP                       4
                                      Type
  INTSEC          1          1        Interval                  7    0          5    0
                                      Seconds
  XICT01SECS      1          2        486 CPU                   7    1
                                      Seconds
  CPU486INT       2          2            486 CPU               7    1          4    1
                                        Utilization
                                            (%)
```

Figure 62 (Part 1 of 2). QAPMIOPD File Query Definition - CPU

```
Selected output attributes
  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


                                   IBM Query/400                    3/22/96   13:59:07        Page    3
Printer Output
  Printer device  . . . . . . . . . . . . *PRINT
  Report size
    Length  . . . . . . . . . . . . . . .  66
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6
  Report end line . . . . . . . . . . . .  60
  Report line spacing . . . . . . . . . . Triple space
  Print definition  . . . . . . . . . . . Yes

Printer Spooled Output
  Spool the output  . . . . . . . . . . . Yes
  Form type . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Copies  . . . . . . . . . . . . . . . .   1
  Hold  . . . . . . . . . . . . . . . . . Yes

Cover Page
  Print cover page  . . . . . . . . . . . Yes
    Cover page title
      8 Users Interconnected - Integrated PC Server CPU

Page headings and footings
  Print standard page heading . . . . . . Yes
    Page heading
      8 Interconnected LAN Server/400 Clients - Integrated PC Server CPU
    Page footing
```

Figure 62 (Part 2 of 2). QAPMIOPD File Query Definition - CPU

```
                                          8 Users Interconnected - Integrated PC Server CPU
                                  QUERY NAME . . . . . CPU486COOK
                                  LIBRARY NAME . . . . DJOHNSON
                                  FILE         LIBRARY      MEMBER       FORMAT
                                  QAPMIOPD     FSCITY2      BAP8WS64MB   QAPMIOXR
                                  DATE . . . . . . . . 03/22/96
                                  TIME . . . . . . . . 13:59:07
                                        Integrated PC Server 486 CPU Utilizationt Statistics


03/22/96  13:59:07  8 Interconnected LAN Server/400 Clients - Integrated PC Server CPU  PAGE    1
Int IOP  Data    IOP    Interval         486 CPU       486 CPU
#   Bus  Type    Type   Seconds          Seconds     Utilization
         (OS/2)                                          (%)
 1    2    2     6506        298           93.8        31.4
 2    2    2     6506        296          131.4        44.4
 3    2    2     6506        299          120.6        40.3
 4    2    2     6506        197           47.1        23.9
                FINAL TOTALS
                TOTAL      1,090          392.9
                AVG                                    35.0
* * *  E N D  O F  R E P O R T  * * *
```

Figure 63. QAPMIOPD File Query Report - CPU

13.6 Multimedia File Serving using USF/400

This section provides performance information for multimedia file serving using V3R6 and Ultimedia System Facilities (USF).

USF is a feature of OS/400 that brings multimedia function to the AS/400 system. Image, audio, and motion video can easily be added to new or existing applications executing in either the client or AS/400 host/server using the provided APIs. Repository and object services support provides the capability to capture, register, and play back multimedia objects such as video clips, audio files, and still images. A repository is provided to catalog and track objects.

Additional capabilities to edit, transform, and sequence objects give a complete support platform on which to build multimedia enabled applications. USF also extends the supersedes the capabilities covered by the Ultimedia Video Delivery System/400.

The USF product requires the use of a PC client running either OS/2 2.1 with Multimedia Presentations Manager/2 or DOS 5.0 and MS Windows 3.1 with Multimedia Extensions. AS/400 host-based applications or PC-based C language applications can make use of the provided APIs. Client Access/400 is required and additional multimedia hardware and software may also be required depending upon the type of multimedia data to be used.

The following workloads were used to evaluate multimedia file serving performance. **Workload Descriptions:**

1. Heavy image:
   A USF sequence object displays a bitmap image every 15 seconds (150KB average size).
2. Heavy image with audio:
   A USF sequence object displays a bitmap image every 15 seconds. An audio clip (16 bit 22Khz) plays continuously in the background while the images (150KB average size) are displaying.
3. Video (high) - continuous high quality DVI video:
   A sequence object repeatedly invokes a production level DVI video file that requires 180KB/sec average transfer rate (this constitutes a high quality DVI file). Files are varied to ensure that file accesses are from disk.
4. Video (medium) - continuous medium quality DVI video:
   A sequence object repeatedly invokes a DVI video file that requires 150KB/sec average transfer rate (this constitutes a medium quality DVI file). Files are varied to ensure that file accesses are from disk.

The following configurations were used for the multimedia tests described in this section.

Configurations:

- Server: 9406 Model 510-2143 running V3R6:
  - 256MB memory, 2-6606 (1967MB) and 1-6602 (1031MB) DASD
  - 2619 token-ring LAN IOP, used at 16 Mbps
  - Frame size = 16393, RU size = 16384
- Client: PS/2 Model 95 - 50Mhz:
  - Tested with both WARP and Windows 3.1
  - 32MB memory
  - 4/16 Mbps token-ring adapter, used at 16 Mbps
  - Action Media II adapter

Measurement Results:

The Optimized for OS/2 Client Access/400 client was used to collect the following information. The following measurement data was reported by the OS/400 performance monitor.

Table 30. V3R6 Multimedia File Serving Performance Data

Ultimedia System Facilities
Server: AS/400 510-2143 V3R6
Client: Optimized for OS/2 CA/400 Running USF

Scenario	**AS/400 CPU Util (%)**	**AS/400 Disk IOP Util (%)**	**AS/400 Disk Util (%)**	**AS/400 TRLAN 2619 IOP Util (%)**	**16 Mbps TRLAN Util (%)**
Heavy Image	1.0	0.9	1.7	4.9	< 1
Heavy Image with Audio	2.3	1.7	2.7	7.2	2.0
Video (high)	2.3	3.3	5.4	8.4	8.0
Video (medium)	2.1	3.6	4.9	7.8	7.0

Note:
The preceding utilizations were generated using a single client.

13.6.1 USF/400 Performance Analysis and Tips

- The results from Table 30 give an indication of how AS/400 system resources are used for a single client executing the various scenarios described.
- Using the Action Media II adapter improved the video quality. This adapter may no longer be available, but other digital video cards can be used to provide hardware assist for off-loading decompression.
- The USF product is built on the QDLS file system and uses QDLS to store object and attribute information for all USF objects. The Integrated File System introduced in V3R1 provides improved performance to the new Client Access/400 clients, and USF allows for files to be served from the Root file system as well as QDLS. Performance improvements can be realized by serving video files from the Root file system. Refer to Section 13.4, “Client Access/400 File Serving Performance” on page 331 for additional information.
- Using the largest frame and RU sizes possible is a necessity when running multimedia applications. 16K frames and RU sizes were used for all measurements. The following steps can be used to adjust the settings for the indicated client:
  - **Optimized for OS/2 client:**
    - click on the AS/400 icon on the desktop
    - click on the CA/400 Connections icon
    - select AS/400 server
    - click on SNA Configuration
    - click on LAN
    - click on Advanced Options
    - set maximum I field size to 16393
    - click on Mode
    - select QPCSUPP and click on change
    - click on Max RU size and enter 16384
    - click on OK, close, and save settings.
  - **Windows 3.1 client:**

        - Click on CA configuration
        - click on System
        - select AS/400 server
        - click on Global Options
        - set maximum frame size to 16393.

Performance tips:

- ″Casual video″ workloads such as non-moving video displays achieve acceptable performance when the image data is stored in QDLS.
- When QDLS and Client Access/400 file serving is being used, consider setting the QPWFSERVER class description to priority 15 while running USF/400 video display functions.

  Note that using Client Access/400 and QDLS for file serving may yield high AS/400 CPU utilization and disk I/O rates that need to be balanced against other work being done concurrently on the system.
- Placing the objects in the Integrated PC Server (LAN Server/400 required) is required in ″heavy video″ workloads, such as displaying a moving video.

  The Integrated PC Server-LAN Server/400 support is recommended when data rates must be in excess of 200KB per second.
- Recommended LAN configuration parameter values on the AS/400 system include:
  - Line and APPC control unit MAXFRAME = 16393
  - APPC control unit LANMAXOUT = 8
  - APPC control unit LANACCPTY = 3
  - APPC control unit LANACKFRQ = 2
  - Mode QPCSUPP MAXLENRU(*CALC)

  See Section 11.12.4, “LAN Line and Control Unit Parameters” on page 213 for additional AS/400 LAN performance parameter considerations.
- Recommended client configuration parameters include:
  - For OS/2, use no less than 16MB of memory.
  - For DOS Extended environments, use no less than 8MB of memory.
  - Use 16KB RAM for the token-ring LAN card.
  - For Windows in the DOS Extended environment, use:
    - TRMF 16393
    - MCAX 32 or higher

13.7 LANRES/400 Performance Tips

LAN Resource Extension and Services/400 (LANRES/400) provides the following server support for a Novell NetWare client and NetWare server:

- Central administration from an AS/400 display of NetWare server users, data, and access security
- Disk serving:

LAN users can transparently store files on the AS/400 disk storage. A single client can access data on a NetWare server and the AS/400 system at the same time.

- Print serving:

  Any AS/400 system can use LAN attached printers (Host-to-LAN printing) and any LAN user can print data to an AS/400 attached printer (LAN-to-Host printing).

- Data distribution:

  AS/400 data can be distributed to a NetWare server and data can be retrieved from the NetWare server to the AS/400 system. This support can be used to ″save″ and ″restore″ server files. Data files may not be translated between EBCDIC and ASCII, but AS/400 source files can be.

  Automated distribution can be performed through AS/400 CL or REXX programs.

Both TCP/IP and SNA APPC protocols may be used. On the NetWare server, we recommend using the 802.2 LAN protocol support on NetWare release 3.12, 4.0, or later. Performance information in this section is limited to OS/400 work management information using SNA APPC as described in the next topic.

13.7.1 LANRES/400 Work Management

APPC connections were used for the following LANRES/400 subsystem and job information. Use this information for performance measurement of LANRES/400 activity and group LANRES/400 jobs into BEST/1 workloads for capacity planning.

13.7.1.1 LANRES/400 Disk Serving

For disk serving, each LANRES/400 disk server job communicates with only one NetWare server. However, each NetWare server can communicate with more than one LANRES/400 server job. A single LANRES/400 disk server can manage up to 50 disk images.

The same disk image can be made available (read only) to multiple NetWare LANRES servers.

For each LANRES/400 disk server, there are two APPC sessions (a send and a receive session). These jobs run in subsystem QLANRES. Each disk server is started through a Submit Job command to any subsystem.

13.7.1.2 LANRES/400 Print Serving

For Host-to-LAN print, there is an AS/400 ″LANRES/400 submitted job″ that communicates to the server through a LANRES/400 APPC job running in subsystem QLANRES. The LANRES/400 job may be submitted to any subsystem.

For Host-to-LAN print, LANRES/400 supports sending the data to the LAN printer and a ″query status″ function for a user′s AS/400 print job or a print server on the LAN.

For LAN-to-Host print, the NetWare print queue is associated with an AS/400 spooled output queue.

On the AS/400 system, there are two APPC sessions for print serving and these jobs run in subsystem QLANRES.

13.7.1.3 LANRES/400 Subsystem Jobs Example

You can use the information in this section for LANRES/400 performance measurement and group the jobs into BEST/1 workloads for capacity planning.

```
QLANRES Subsystem

   Job Name     User         Number   Type      -----Status-----

   QLANAPPC     QLANRES      128972   BATCH     ACTIVE
                                  1
   QLANAPPC     QLANRES      128974   BATCH     ACTIVE
   QLANAPPC     QLANRES      128978   BATCH     ACTIVE
   QLANAPPC     QLANRES      128984   BATCH     ACTIVE
   QLANAPPC     QLANRES      128987   BATCH     ACTIVE
   QLANAPPC     QLANRES      128990   BATCH     ACTIVE
   QLANAPPC     QLANRES      128993   BATCH     ACTIVE
   QLANAPPC     QLANRES      128996   BATCH     ACTIVE
   QLANAPPC     QLANRES      128999   BATCH     ACTIVE
   QLANAPPR     QLANRES      128973   BATCH     ACTIVE
                                  1
   QLANAPPR     QLANRES      128975   BATCH     ACTIVE
   QLANAPPR     QLANRES      128979   BATCH     ACTIVE
   QLANAPPR     QLANRES      128985   BATCH     ACTIVE
   QLANAPPR     QLANRES      128988   BATCH     ACTIVE
   QLANAPPR     QLANRES      128991   BATCH     ACTIVE
   QLANAPPR     QLANRES      128994   BATCH     ACTIVE
   QLANAPPR     QLANRES      128997   BATCH     ACTIVE
   QLANAPPR     QLANRES      129000   BATCH     ACTIVE
   QLANCTLR     QLANRES      128965   AUTO      ACTIVE
                                  2


xxxx Subsystem

   Job Name     User         Number   Type       LANRES
                                                 Function
    HL01Q       QSECOFR      129958     B     Host->Lan Prt
                                   3
```

Figure 64. LANRES/400 Subsystem Jobs

Figure 64 shows the APPC send/receive pair jobs in the QLANRES subsystem. **1** are probably the send/receive pairs for the same NetWare server. QLANCTLR **2** is an autostart job that provides control functions. You need to DSPJOB and display lock to verify that the APPC device description for an attached server is being processed by each **1** job.

The HL010 job **3** was submitted by the AS/400 LANRES operator and is a host-to-LAN print job. The name ″HL010″ was assigned by the user as a meaningful name to identify this job from other jobs that may be running in the same subsystem. For example, ″010″ represents a specific NetWare server print queue.

We recommend running the jobs in QLANRES subsystem and the LANRES/400 function jobs in their own storage pool. You may also consider changing their run priority based on LANRES/400 requirements and other applications running at the same time.

13.8 Novell NetWare for OS/400

Novell NetWare for OS/400 is not available in version V3R6 but will be available in V3R7; the current product is available in V3R1 and will be available in V3R2.

As your business computing environment grows more complex, your need to manage that environment increases. You may need to:

- Manage local area networks ***(LAN)*** separately.
- Manage differences between applications and systems.
- Find more efficient and reliable methods of saving your local area network (LAN) data.
- Manage data resources by consolidating data.

The OS/400 integration for Novell NetWare feature allows you to do these tasks by installing and running Novell NetWare on a File Server I/O Processor within your AS/400 system.

OS/400 Integration for Novell NetWare uses OS/2 WARP to install and run NetWare 4.1. In the installation process, several server storage spaces are created that are used by OS/2 Warp to load OS/2. These storage spaces are also used for the installation and startup of the NetWare server. The OS/2 drives: **C:, D:, E:, and F:** are formatted as FAT disks.

The C: drive also contains the **NET.CFG** file that contains configuration parameters for NetWare for OS/2. One of the main parameters in this file is PERFORMANCE TUNING. This parameter is used to set the division of the processor between OS/2 and the NetWare server. A setting of 10 gives the highest possible processor time to the server. The default setting is 8, which makes the assumption that most of the Integrated PC Server processing power is dedicated to the NetWare server.

You cannot edit the ***NET.CFG*** file.

An easy way to change the PERFORMANCE TUNING value between OS/2 and NetWare is to add the following line in your AUTOEXEC.NCF file:

```
SET PERFORMANCE TUNING n
```

Where **n** is a value from 1 to 10.

13.9 ADSM/400 Performance Tips

ADSM/400 provides rich support for backing up (saving) and restoring files and directories of IBM and OEM clients attached to the AS/400 system. Clients running the appropriate ADSM client support include the following operating systems:

- DOS
- IBM OS/2 with LOTUS NOTES Agent
- Novell NetWare
- Apple Macintosh OS
- Microsoft Windows
- Microsoft Windows NT
- AT&T Global Information Solutions

- IBM AIX
- DEC ULTRIX
- Hewlett-Packard HP-UX
- NEC EWS-UX/V
- SCO UNIX 386/SCO Open Desktop
- Sequent PTX
- Siemens-Nixdorf SINIX
- Silicon Graphics IRIX
- Sun Microsystems SunOS/Solaris
- IBM OpenEdition MVS

Backup can be automated by ADSM/400 configuration parameters or upon request from a client and either APPC (CPI-C interface) or TCP/IP protocols may be used. The redbook, *Setting Up and Implementing ADSTAR Distributed Storage Manager/400*, GG24-4460, and *Performance Capabilities Reference*, ZC41-8166, provide a detailed description of ADSM/400 configuration and client configuration parameters and performance test results that are beyond the scope of this performance management redbook.

This topic provides summary level information on general data rate expectations, configuration parameters that affect performance, and some OS/400 work management examples that can be used to group ADSM work for performance measurement and capacity planning on a customer system.

Key performance tips (many shared with most client/server applications) include:

- Use the fastest LAN speed and largest frame sizes where possible.
- For SNA APPC connections, use a large RU size and pacing value.
- Backing up or restoring with AS/400 ″memory/disk storage″ is faster than going directly between the client and a tape device.
- The initial backup of a client is a full backup and takes considerable time and system resources. After the initial backup, we recommend a ″save changed files only″ approach.
- Run the heavy periods of client activity when there is little other system work.
- Backup of a single large file is significantly faster than backing up several small files that consume the same amount of total storage. For example, a single 25MB file backup is significantly faster than an entire 25MB directory made of 100 files within several subdirectories.
- When implementing a client backup approach, start out running five or less clients within a time period while collecting Performance Monitor data. Use the information contained in Section 13.9.2, “ADSM/400 Work Management” on page 367 to observer AS/400 resource utilization and group ADSM jobs into BEST/1 for modeling system resources utilization of 10, 20, and so on clients and consider multiple LAN adapters on the AS/400 system.
- Use ADSM/400 parameters to limit the number of attached clients doing backup/restore during the same time period.

  This is specified under the ADSM/400 ″MAXSCHedsessions″ (maximum scheduled sessions) parameter. A randomizing factor (″RANDomize″) can also be specified to ensure all scheduled sessions do not start during the same few seconds. ″QUERYSCHedperiod″ specifies when clients can query ADSM/400 to obtain scheduled work.

- Data compression considerations include:
  - Compression may significantly reduce server (AS/400) storage consumption.
  - Compression can minimize the exposure to the LAN becoming a bottleneck during peak data exchange.
  - Compression reduces the CPU utilization on the server.
  - Compression increases the total elapsed transmission time for an individual client.
  - Compression increases client processor utilization.
  - Compression may increase total backup throughput when multiple clients are active within the same time period (window):

    If the clients have sufficient compression performance, the reduction in server CPU utilization may result in decreased total run time for all backups.
- The faster the client processor, the better throughput performance.

Refer to the following documents on the marketing tools disk for the most current information relating to ADSM server and client performance.

Table 31. ADSM/400: Performance Documents

Document	**Description**
ADSM400	ADSM/400 Server Performance Measurements Release 2
ADSM21TG	ADSM Performance Tuning Guide, Version 2 Release 1
ADSMCAP	ADSM Capacity Planning Tool

13.9.1 ADSM/400 AS/400 APPC and TCP/IP Configuration

The *Performance Capabilities Reference V3R2*, ZC41-8166, contains many performance test results based on the various AS/400 system and client parameter values. The following values are presented as good ″starting values″ for ″best″ ADSM/400 V3R6 performance.

13.9.1.1 ADSM/400 SNA and TCP/IP Parameters

The following values are recommended initial parameter values for token-ring LAN, APPC, and TCP/IP parameters:

- TRLAN I Frame (MAXFRAME) = 16393
- TRLAN Early Token Release = *YES
- TRLAN logging *OFF
- TRLAN Send Window Count = 2
- TRLAN Receive Window Count = 1
- SNA Mode (#ADSM) Receive Pacing = 63
- SNA Mode (#ADSM) MAXLENRU = 16384
- CPI-C Buffer Size = 31
- OS/2 dsm.opt, AIX dsm.sys CPIC Buffer Size = 31
- OS/2 dsm.opt, AIX dsm.sys CPIC Mode Name = #ADSM

13.9.1.2 ADSM/400 through TCP/IP

- TCP Keep Alive = 120
- TCP Urgent Pointer = *BSD
- TCP Receive Buffer Size = 16384
- TCP Send Buffer Size = 16384
- UDP Checksum = *NO
- IP Datagram Forwarding = *YES
- IP Reassembly Timeout = 120
- IP Time to Live = 64
- ARP cache timeout=5
- Log Protocol Errors = *NO
- Uncompressed data:
  - OS/2 dsm.opt, AIX dsm.sys TCPBuffersize = 16
  - OS/2 dsm.opt, AIX dsm.sys TCPWindowsize = 8
- Compressed data:
  - OS/2 dsm.opt, AIX dsm.sys TCPBuffersize = 5
  - OS/2 dsm.opt, AIX dsm.sys TCPWindowsize = 2

13.9.2 ADSM/400 Work Management

You need to understand that jobs in subsystem QADSM and the jobs that actually send or receive the data work together. Subsystem QADSM jobs do most of the internal space allocation, backup scheduling, and ″record keeping″ for attached clients and their files. Note that ADSM/400 uses several database files and stores the client data in an internal ADSM/400 format that is not defined externally. ADSM/400 transactions can be logged within one of ADSM′s database files.

ADSM/400 code runs from library QSYS and typically, user configurations, data, and transaction log information are kept in library QUSRADSM.

Subsystem QCMN (for APPC) or subsystem QSYSWRK (for TCP/IP) jobs perform the actual exchange of data between the AS/400 system and the client.

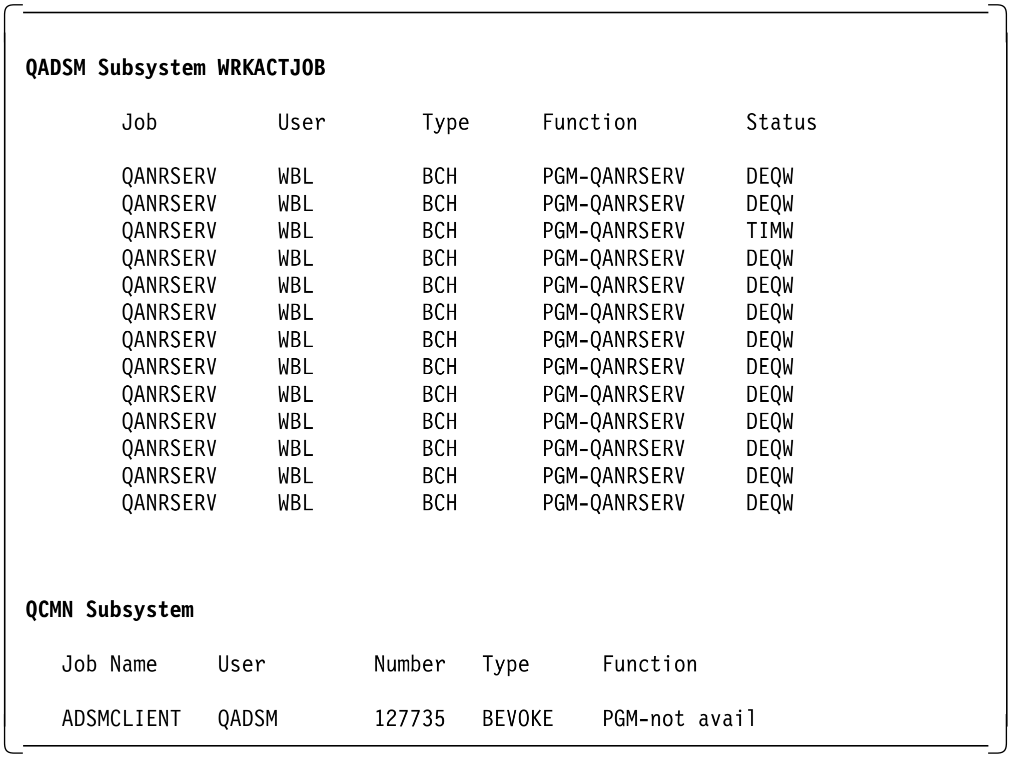
When using APPC, it is strongly recommended you create an AS/400 mode description (CRTMODD) ″QADSM″ for ADSM/400 functions and add a communications entry (ADDCMNE) for the client APPC device description to control QADSM job security, run priority, and storage pool allocation. The following example is recommended:

```
o ADDRTGE SBSD(QCMN) SEQNBR(300) PGM(*RTGDTA) CMPVAL(′QADSM′ 1)
          CLS(QGPL/QBATCH or user-defined)

o ADDCMNE SBSD(QCMN) DEV(*APPC) JOBD(*USRPRF) DFTUSR(QADSM)
          MODE(QADSM)
```

Figure 65 on page 368 represents the QADSM jobs and QCMN jobs that are active during a client backup or restore. The jobs in QADSM remain active in a dequeue wait (DEQW) status or timed wait (TIMW) status when no work is being performed.

Note that the Client ADSM function provides a real-time snapshot of the data transfer rate in kilobytes per second. This is actually only a snapshot of the transfer rate taken every few seconds. In general, the rate (or rates) shown indicates much higher transfer rates than actually occur.

The greater the number of separate files sent, the greater the difference between your actual results and the rates shown on the ADSM/400 client display.



```
QADSM Subsystem WRKACTJOB

        Job          User        Type      Function         Status

        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     TIMW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW
        QANRSERV     WBL         BCH       PGM-QANRSERV     DEQW



QCMN Subsystem

   Job Name     User         Number   Type      Function

   ADSMCLIENT   QADSM        127735   BEVOKE    PGM-not avail
```

Figure 65. ADSM/400 Subsystem Jobs Example

This figure shows the QADSM subsystem jobs that are available to perform various ADSM/400 functions and the QCMN job for a specific client workstation that performs the actual data exchange between the client and the AS/400 ADSM Server.

All QADSM subsystem job names are identical (no definition of their specific function). You may collect Performance Monitor data during backup, restore, and other functions for these jobs and the QCMN client job to measure performance and build a BEST/1 workload for capacity planning.

All QADSM jobs run under the user ID of the user who issued the Start ADSM Server function. Display class description QADSM to determine run priority as shipped.

Subsystem QADSM is shipped with storage pool 1 - *BASE. If you plan on running ADSM/400 functions concurrently with other applications, consider defining and using a specific storage pool other than *BASE.

The QCMN job shown (ADSMCLIENT QADSM 128735) is the APPC (device description) job associated with the client that actually transfers the data. The job must run under profile QADSM as currently required by ADSM/400 implementation.

If the AS/400 administrator for ADSM/400 uses the default communications entry and routing entries for subsystem QCMN, the data transfer job runs at priority 20 using mode #INTER which defaults to class QINTER. If you have many clients exchanging large files/directories at this priority, you may impact already scheduled interactive or batch night jobs. You need to consider this when setting up times for ADSM/400 functions. Therefore, for QCMN, a recommended routing entry sequence for mode QADSM is as follows:

```
 Seq Nbr    Program    Compare Value          Start Position

  290      *RTGDTA     ′QPCSUPP′                       1
  295      *RTGDTA     ′QCASERVR′                      1
  299      *RTGDTA     ′#INTER′                        1
  300      *RTGDTA     ′QADSM′   <-- Class QBATCH      1
  310      *RTGDTA     ′PGMEVOKE′                     29
```

13.10 DataPropagator Relational/400

DataPropagator for the AS/400 system joins the IBM family of data replication products used to move data among heterogeneous relational and non-relational databases. It provides the interface for copying data in a variety of ways (summarized, net changes, snap shots) among systems. It also handles the application of that information to selected target databases on remote systems. This is a powerful tool for customers building distributed client/server applications.

Scenarios in which DataPropagator Relational/400 are appropriate include:

- Periodic replication of production data from a large AS/400 system to a server model AS/400 system for use by Decision Support users, perhaps in a GUI environment.
- Distribution of subset information to AS/400 systems at remote locations.
- Consolidation of corporate data at a central site from multiple locations.

The replication process can select records based on selected values, summarize the data, extract specific columns/fields rather than copying an entire record, and initiate tasks on the source or target systems before or after the replication process.

Customers do this type of application interface today, but typically build custom application code to satisfy each new replication need. DataPropagator Relational/400 can now greatly reduce the effort required for data replication. Users simply register databases as available for replication and subscribe for replication services at other sites. DataPropagator Relational/400 guarantees safe delivery of the data.

As a general rule, DataPropagator Relational/400 is not a solution for those customers who need hot-site backup of production data.

For V3R6, DataPropagator/400 has the following increased capabilities:

- Ability to replicate non-keyed files:

  An assumption for using the DataPropagator product is that the source file contains a column or set of columns that uniquely identifies each record. However, many customers have no such columns that can provide this uniqueness. So, DataPropagator added the capability to specify the Relative Record Number (RRN) of the source file as the unique key. With this support, a new column is added to the target file for the RRN. This is used as the key column.

- Ability to replicate multi-membered files:

  Previously, DataPropagator/400 could be used to propagate from the first member of a multi-membered file. For customers who used such files, there was no solution. Now, DataPropagator/400 can be used to propagate changes from individual members in a multi-membered file. Each member in the source file maps to an individual target file.

- Ability to replicate to existing target files:

  Previously, DataPropagator/400 could be used to propagate to an existing target file because DataPropagator/400 expected to create the target file during the subscription process. In addition, this new target file had an additional column added to the target file that precluded customers from running existing applications. Now DataPropagator/400 can be used to subscribe to an existing target file and create a target file that does not contain additional columns. Customers can use existing applications to access the target files without requiring modification.

Note: All of these capabilities are also available for V3R1 by ordering the following 5763DP1 PTFs: SF28005, SF28691, and SF28248.

13.10.1 DataPropagator/400 Work Management

DataPropagator uses the term ″Data Server″ to represent the system where the database changes are being made and the term ″Copy Server″ to represent the system where the changes are *propagated to*.

The Data Server system is where files/tables are registered for propagation. Journals must be defined for the tables that are registered because the changes sent are based on journal receiver entries.

The capture process runs on the Data Server (″source system″). It processes journal entries for the files for which change data is required and updates a table that is used as a staging table for the apply process (or processes) on the Copy Server systems.

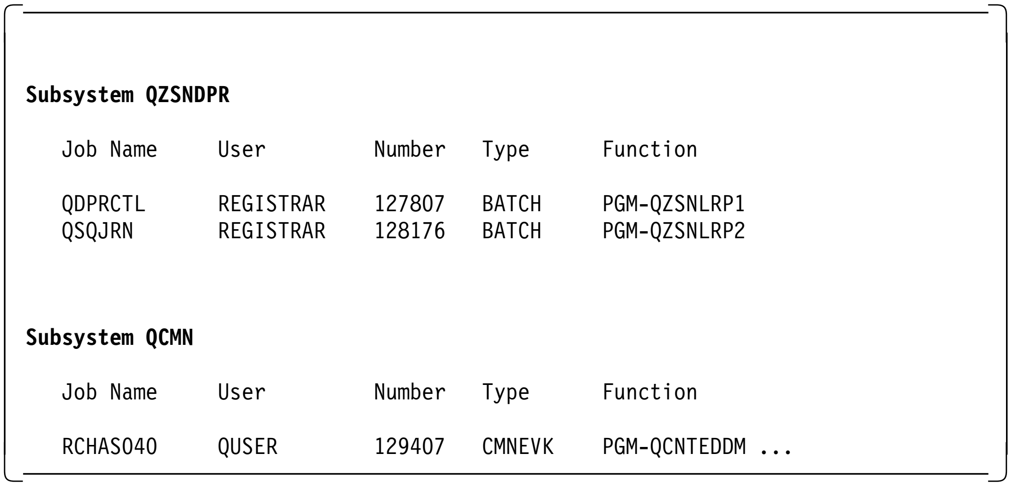
The Copy Server system defines its DRDB parameters and the frequency of requests for changes. It is the frequency of these requests and the complexity of the SQL statements that can impact CPU utilization on both the Data Server system and the Copy Server system.

The apply process runs on the Copy Server (″target system″). Using DRDA, the apply process either applies changes to the target file (using the staging table from the Data Server as input) or it may refresh the target file completely, using the original source table as input. The user can determine whether copies are applied or the table is refreshed at the specified intervals.

The following topics provide an example of the subsystems and subsystem jobs that run when the capture and apply processes are active on each system. Use this information when doing performance measurement and capacity planning (BEST/1) of DataPropagator support.

13.10.1.1 Data Server Subsystems

For the Data Server system, the jobs QDPRCTL and QSQJRN perform the registration management and data capture processes where QDPRCTL is the controlling job and QSQJRN is the actual capture process job. QSQJRN is the name of this job because the job name takes on the name of the journal it is capturing changes from. The QCMN job is the DDM conversation job that was started by the Copy Server system when it requested the changes be sent to it.



```
Subsystem QZSNDPR

   Job Name     User         Number   Type      Function

   QDPRCTL      REGISTRAR    127807   BATCH     PGM-QZSNLRP1
   QSQJRN       REGISTRAR    128176   BATCH     PGM-QZSNLRP2


Subsystem QCMN

   Job Name     User         Number   Type      Function

   RCHAS040     QUSER        129407   CMNEVK    PGM-QCNTEDDM ...
```

Figure 66. DataPropagator/400 - Data Server Subsystem Jobs

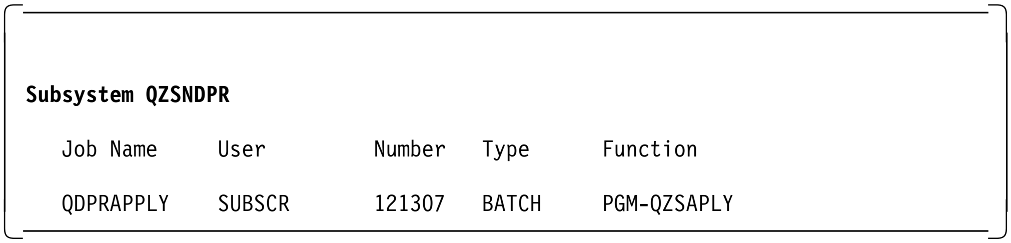
The user ID for the QCMN job is QUSER because the Copy server is not defined as a secure location to the Data Server. When the DRDA communication is established, the default user for the QCMN subsystem is used instead of user ID SUBSCR which started the Apply process.

Performance Tuning Suggestions

- If your propagation needs are daily or greater, consider keeping Capture off (but journaling remains on) during peak hours. Start Capture after peak hours when CPU availability is greater.
- Capturing the entire record (versus selected fields) with after images only (versus both before AND after images). This is determined at registration time.
- Lengthening the Apply interval of the subscriptions can also reduce the CPU overhead on the data server.
- Manage pruning time so that pruning is not done during peak hours.
- Changing the job priority to a higher priority for the QSQJRN job reduces the capture process time. However, this may affect other applications running on the system.

13.10.1.2 Copy Server Subsystems

Subsystem QZSNDPR must be running on the Copy Server system as well as the Data Server system. On this Copy Server system, the QDPRAPPLY job applies the changes received through DDM job 129407 on the Data Server system.



```
Subsystem QZSNDPR

   Job Name     User         Number   Type      Function

   QDPRAPPLY    SUBSCR       121307   BATCH     PGM-QZSAPLY
```

Figure 67. DataPropagator/400 - Copy Server Subsystem Jobs

Note: You can configure both the Data Server and the Copy Server functions on the same system. This technique is useful for testing and capacity planning purposes.

Performance Tuning Suggestions

- Lengthening the Apply intervals, particularly if there are a large number of subscriptions, allows the Apply process to process fewer large blocks of changes instead of many small blocks of changes.
- Splitting the subscriptions among several user IDs allows several Apply processes to run the subscriptions in parallel rather than sequentially.
- Reducing the amount of resources used for full refresh, especially for large files, can be accomplished by following the procedure suggested in Appendix E of the *DataPropagator Relational Capture and Apply/400*, SC41-3346.
- Changing the job priority to a higher priority for the QDPRAPPLY job can reduce the Apply process time. However, this may affect other applications running on the system.

13.11 OptiConnect/400

OptiConnect/400, 5716-SS1 (option 23) is a new part of OS/400 V3R6. The AS/400 Model 530 offers customers a ″vertical″ growth path consisting of bigger, more powerful processors. For those customers environments where a single AS/400 model is not sufficient, OptiConnect/400 offers customers a complementary ″horizontal″ growth path by enabling from two to 32 high-end AS/400 systems to have a transparent access to data using high-speed fiber optic bus connections and an enhanced version of AS/400 Distributed Data Management (DDM).

13.11.1 OptiConnect/400 Concepts

By using OptiConnect/400, customers can increase their system capacity by distributing their application and database workload across multiple AS/400 systems.

An AS/400 system containing a central database can ″serve″ many applications running on multiple ″client″ systems, allowing customers to grow their computing resources while maintaining a single database.

OptiConnect/400 can also provide significant CPU savings for existing DDM environments.

While the database client/server relationship is not new to the AS/400 architecture, what is new is the use of fiber optic technology to provide high-speed transparent database access across systems.

There are two main components to this solution:

1. Hardware:

   The AS/400 optical bus technology allows system-to-system connection of two speeds for the RISC processor: 266 Mbps and 1063 Mbps. When running any speed, both sides must comply with the same speed connection. 266 Mbps requires a #2683 feature in one AS/400 system and a #2686 feature on the other AS/400 system. For the 1063 Mbps optical connection, a #2685 feature is required on one AS/400 system and a #2688 feature is required on the other AS/400 system. You cannot mix the cards. For example, a #2685 feature cannot be connected to #2686 feature.

2. Software:

   The AS/400 system has an existing transparent database access solution in the form of Distributed Data Management (DDM). OptiConnect/400 uses a combination of hardware and software technologies to deliver an enhanced subset of DDM requests across the optically connected client and database server systems.

With OptiConnect/400, now we have the ability to isolate the database function on one or more separate CPUs. Single system database integrity is maintained and only a small response time increment is added to application database access.

OptiConnect/400 installations are non-trivial. The OptiConnect/400 installation process requires active customer participation and on-going management and support.

13.11.2 OptiConnect/400 Performance Considerations

OptiConnect/400 (OC/400) enables applications to transparently access databases on remote AS/400 systems. It allows applications written to access databases locally, to access them remotely with simple changes to file descriptions, and no changes to the applications.

OC/400 provides an optimized DDM solution capable of efficient, low-latency, high bandwidth communication between AS/400 systems. OC/400 utilizes a fiber optic connected shared I/O bus and a SLIC shared bus device driver to transport data and messages. The efficiency of OC/400 allows horizontal growth solutions (increasing total capacity to shared data beyond the bounds of a single system). Prior to OC/400, DDM using ICF/APPC for transport has provided the remote data access function but the overhead of communication between systems was too high to achieve horizontal growth under heavy workloads. This overhead kept DDM from being a viable solution for user′s horizontal growth. OC/400 offers customers this horizontal growth by enabling two to 32 AS/400 systems to share an I/O bus. Any or all systems can act as both an application machine (workstation server) for accessing another system′s data or a database server providing data to other systems. OC/400 specialists in Rochester can determine the horizontal growth feasibility. For more information, please contact the OC/400 solution manager in the U.S. at 1-800-426-4260. For EMEA, please contact the OC/400 solution manager in the U.K. at (44) 181 681 4419.

Table 32 on page 374 shows system capacity growth potential when OC/400 is used in system environments with various levels of database activity. This table only applies to the work that is being considered for OC/400 usage. The database server machine is assumed to have similar configuration such as the single system has. These are only guidelines and results vary depending on system environments.

Table 32. OC/400 Growth

OC/400 Horizontal Growth Potential
Single Database Server and Multiple Application Machines

DB %	1-C	2-C	3-C	4-C	5-C	6-C	7-C
10	1.81	2.63	3.44	4.26	5.07	5.89	6.70
20	1.66	2.31	2.97	3.62	4.17	-	-
30	1.52	2.03	2.55	2.78	-	-	-
40	1.39	1.79	2.08	-	-	-	-
50	1.29	1.57	1.67	-	-	-	-
60	1.19	1.38	1.39	-	-	-	-

Note:

- DB% = The percent of one-system CPU time allocated to database activity.
- 1-C = two systems composed of one application machine and one database server.
- 2-C = three systems composed of two application machines and one database server.
- Client Factor = OC/400 COMM CPU associated with application machine.
- Server Factor = OC/400 COMM CPU associated with database server.
- In this table, a client factor of 1.8 and a server factor of 1.2 is used.
- A typical customer environment includes one database server and two or more application machines.
- The columns show the capacity gain over a single system. For example, in an environment that 20% of CPU is consumed for database activity, column 2-c shows a value of 2.31. This value is the capacity growth achieved by using three systems; one as server and two as application machines over a single system.
- All systems are assumed to have the same CPW value for this exercise.
- Database server system has similar hardware configuration such as the single system does.

13.11.2.1 Additional Performance Considerations for OptiConnect/400

The following conclusions are derived from laboratory benchmarks and from experience gained working with IBM OptiConnect/400 customers:

1. Each time that a ″put″ or ″get″ is issued, OC/400 uses the shared I/O bus to exchange data with the remote system. DDM uses a communication link for accessing data on a remote system.
2. V3R6 has similar or better performance to V3R1 for local and OC/400 when transaction capacity is scaled to CPW of the CPU.
3. OC/400 is a more efficient way than DDM to access data on another AS/400 system.
4. At higher levels of throughput, the system overhead for OC/400 increases in a non-linear fashion. This is due to multiprocessing contention when CPU utilization is higher. It is recommended that CPU utilization should not exceed 90% on either the application machine or database server system for this reason.

5. There is an overhead (called client/server factor) associated with using OC/400 both on the application and database machine. This overhead varies and is dependent on factors such as:

   - Percent of database activity
   - Number of logical I/Os per transaction
   - Usage of Journal and Commitment Control
   - Number of database Open/Close

6. The Maximum length for the OC/400 fiber optic cables to the shared bus is 2KM, requiring all OC/400 systems to be within 2KM of the system providing the shared bus. Due to fiber latency of five nano-seconds per meter, it is recommended that systems should be within 300 meters of each other. This may avoid any performance impact due to line latency. DDM using APPC over LANs or WANs can provide greater distances.

7. The OC/400 horizontal growth is achieved across multiple systems with some or all systems acting as servers. Any single database network of logical and physical files must reside on a single system. Therefore, multiple servers requires multiple, separate database networks. The maximum load supported across multiple systems is typically limited by the capacity of a database server (additional application systems can be added until the database server saturates).

8. When analyzing batch workloads, more attention must be paid to logical database I/O′s, blocking, and pool size. Performance characteristics may be different for batch versus interactive use of OC/400. In particular, batch jobs are often database intensive and accumulated CPU and delays for each logical I/O that must traverse the OC/400 shared bus can add significant CPU utilization and elapsed time to batch runs.

9. A detailed analysis to determine where to place the data files is critical in batch workloads or read-only files.

10. An OC/400 shared bus adapter can support up to 40MB per second data rate. Measured bus/adapter utilization in the majority of OC/400 testing has shown CPU limits are reached prior to reaching bus/adapter limits (that is, 100% CPU with <2MB/sec bus usage). Multiple jobs are required to maximize bus utilization.

11. In most distributed environments, including OC/400, there is a penalty for remote access; obviously this penalty is most severe when 100% of accesses are remote and may become negligible for lower percentages (for example, 10%). Therefore, an optimum environment should minimize remote access by distributing users or data across systems in a manner to maximize local access. Replication of read-only files to application machines, distribution of data across multiple database servers, and location of batch work on application machines or database servers must be carefully considered when defining an OC/400 environment to optimize total system capacity.

13.11.3 What OptiConnect/400 Is Not....

If you plan to use OptiConnect/400 as a means of horizontal system growth, be aware that Opticonnect/400 is not a substitute for:

System Clustering:

OptiConnnect/400 is not ″clustering″.

Many industry related descriptions of ″clustering″ point to the transparent and seamless merging of resources. For example, four ″uni″ processors clustered together have a single system image of a four-way processor. All of the systems in the cluster are able to share memory, DASD, and so on.

Shared DASD:

OptiConnect/400 is not ″shared DASD″. While OptiConnect/400 enables transparent database access, the database server owns the database that is on its processor. As in any resource sharing technology, the availability of the complex is based on the availability of the common resources. In the case of OptiConnect/400, the database server is the shared resource and special consideration should be given to ensuring its availability.

Single System Image:

Although OptiConnect/400 presents a single system image of the database from an application standpoint, it is not a single system image from an operations department or system′s management viewpoint. Standard operating procedures and management policies continue to apply to all AS/400 systems that use Opticonnect/400.

Chapter 14. Internet Connection for AS/400 System

Performance information for the Internet Connection for AS/400 product (part of OS/400 V3R7) is included in this section. There are many factors that can impact overall performance (for example, end-user response time, throughput) in the complex Internet environment, some of which are shown in the following list:

- Web Browser:
  - Processing speed of the client system
  - Performance characteristics of the Web browser
  - Client application performance characteristics
- Communications network:
  - Speed of the communications links
  - Capacity of any proxy servers
  - Congestion of network resources
- AS/400 WWW (Web) server:
  - AS/400 processor speed
  - Utilization of key AS/400 resources (CPU, IOP, memory, disk)
  - Web server performance characteristics
  - Application performance characteristics

The primary focus of this section is to discuss the performance characteristics of the AS/400 system as a server in a Web serving environment, providing capacity planning information, and recommendations for best performance. Please refer to Section 11.12, "Communication Performance Considerations" on page 209 for related information.

Data accesses across the Internet are distinctly different from accesses across ″traditional″ communications networks. The additional resources to support Internet transactions such as CPU, IOP, and line speed are significant and must be considered in capacity planning. Typically, in a traditional network:

- Request and response (between client and server).
- Connections/sessions are maintained between transactions.
- Networks are tuned to use large frames.

For Internet transactions, there are a dozen or more line transmissions (including acknowledgements) per transaction:

- A connection is established between client and server.
- Request and response (between client and server).
- The connection is closed.
- Networks typically have small frame (MTU) sizes.
- HTML windows to browser contain more bytes than traditional windows.
- One user transaction may contain several separate internet transactions.

The information that follows is based on performance measurements and analysis done in the AS/400 division laboratory. The highlights, general conclusions, and recommendations documented here do not represent any particular customer environment. Actual performance may vary significantly in a customer environment.

The Hypertext Transfer Protocol (HTTP server) allows AS/400 systems attached to a TCP/IP network to provide objects to any Web browser. At a high level, the connection is made, the request is received and processed, a file system is accessed (or a CGI program is accessed), the data is sent to the browser, and the connection is ended. The HTTP server jobs and the communications router tasks are the primary jobs/tasks involved (there is not a separate user job for each attached user).

Typical HTTP workloads are:

- HTTP: serves a simple HTML page (text only) with HTTP.
- HTML CGI.bin: serves a simple HTML page with CGI.bin in ″C″.
- Net.Data HTML: a simple Net.Data HTML macro.
- DB2WWW HTML: a simple DB2WWW HTML macro.
- SQL CGI.bin: performs a simple SQL request with a CGI.bin in ″C″.
- Net.Data HTML: a simple Net.Data SQL macro that retrieves a table (two rows) and generates a report.

14.1 Web Serving Performance Conclusions/Recommendations

1. **Web Serving Capacity (Example Calculation):** Throughput for web serving is typically discussed in terms of the number of hits/second (connections/second and transactions/second mean the same). Typically, the CPU is the resource that limits capacity. If a large AS/400 model is used with a single LAN IOP, the IOP may be the limiting factor.

2. **PowerPC AS/400 Models and non-PowerPC AS/400 Models:** The HTTP serving environment on PowerPC AS/400 models provides significantly better performance than on non-PowerPC (CISC) AS/400 models with similar CPW ratings. This is due to a number of factors including better performance of the C-language environment and performance improvements to TCP/IP and IC/400. Note that CPW values (relative system performance metrics) are derived from a commercial workload. Therefore, relative capacity ratios for Web serving may not track linearly. However, they should be adequate for a high-level capacity planning exercise.

3. **Response Time (general):** User response time is made up of Web browser (client workstation) time, network time, and server time. Web browser time typically contributes 0.5 seconds to two seconds to response time. A problem in any one of these areas may cause a significant performance problem for an end user. To an end user, it may seem apparent that any performance problem is attributable to the server, even though the problem may lie elsewhere.

   It is common for pages that are being served to have imbedded images (.gif′s and .jpg′s). Each of these separate Internet transactions adds to the response time as they are serially retrieved from various servers (some browsers can concurrently retrieve more than one URL).

4. **HTTP and TCP/IP Configuration Tips:**

   a. The **number of HTTP server jobs** (CHGHTTPA command): Controls the minimum and maximum number of HTTP server jobs handling HTTP requests. The reason for having multiple server jobs is that when one server is waiting for a disk or communications I/O to complete, a different server job can process another user′s request. Also, for N-way systems, each CPU can simultaneously process server jobs. The system adjusts the number of servers that are needed automatically (within the bounds of the minimum and maximum parameters).

The minimum allowed is two (one parent and one child server). Typically three to five servers are adequate for smaller systems. For larger systems dedicated to HTTP serving, increasing the number of servers to 10 or more may provide better performance. Also, if CGI or Net.Data is used (these are more disk I/O intensive), configuring more servers may provide better performance.

   b. The **maximum frame size parameter** (MAXFRAME on LIND) can be increased from 1994 bytes for TRLAN (or other values for other protocols) to its maximum of 16393 bytes to allow for larger transmissions. Generally, documents are much larger than 1994 bytes.

   c. The **maximum transmission unit (MTU) size** parameter (CFGTCP command) for both the route and interface affect the actual size of the line flows. Increasing these values from 576 bytes to a larger size (up to 16388 bytes) probably reduces the overall number of transmissions and, therefore, increases the potential capacity of the CPU and the LAN IOP.

      Similar parameters also exist on the Web browser. The negotiated value is the minimum of the server and browser (and perhaps any bridges/routers), so increase them all.

   d. Increasing the **TCP/IP buffer size** (TCPRCVBUF and TCPSNDBUF on the CHGTCPA command) from 8KB to 64KB may increase the performance when sending larger amounts of data.

5. **HTTP Server Memory Requirements:** Follow the faulting guidelines suggested in Appendix A, "Guidelines for Interpreting Performance Data" on page 383 by observing/adjusting the memory in both the machine pool and the pool that the HTTP servers run in.

   As a guide, based on servers set up in the ITSO Rochester, try using initially the following values for the HTTP server pool:

   ```
   1000KB + 500KB for each HTTP server
   ```

   Factors that may significantly increase the memory requirements include using larger document sizes, using CGI.bin programs, and using Net.Data.

6. **AS/400 Model Selection:** Consult your IBM Systems Engineer or IBM Business Partner to assist in the sizing process. If performance data is available for a given HTTP workload, capacity planning tools such as BEST/1 can be used to determine the appropriate AS/400 model. All of the tasks associated with HTTP serving are ″non-interactive″, so AS/400 server models normally provide the best price/performance.

7. **File System Considerations:** Web serving performance varies significantly based on which file system is used. Each file system has different overheads and performance characteristics. Web serving from the ROOT or QOPENSYS directories generally provide the best system capacity. If Web page development is done from another directory, consider copying the data to a higher-performing file system for production use.

8. **File Size Considerations:** Web serving larger files can use considerably more system resources. The connect and disconnect costs are similar regardless of size, but cost for the transmission of the data with TCP/IP and the IFS access varies with size.

9. **Communications/LAN IOPs:** As there are a dozen or more line flows per transaction, the Web serving environment utilizes the LAN IOP more than other communications environments. Use the performance monitor (STRPFRMON) and the component report (PRTCPTRPT) to measure IOP

utilization. Attempt to keep the average IOP utilization at 60% or less for best performance. IOP capacity depends on file size and MTU size (make sure you increase the maximum MTU size parameter).

On larger AS/400 models, the LAN IOP may become the bottleneck before the CPU does. If additional HTTP capacity is needed, multiple IOPs (with unique IP addresses) can be configured. The overall workload must be ″manually″ balanced by Web browsers requesting documents from a set of interfaces. The load can also be balanced across multiple IP addresses by using a distributed name server.

14.2 5250/HTML Workstation Gateway (WSG)

The 5250/HTML Workstation Gateway enables all Web browsers to be clients to existing 5250 applications without making changes to the application. At a high level, the connection is made, the request is received and processed, the user job processes the application, the workstation I/O is converted from 5250 to HTML, the HTML is sent to the browser, and the connection is ended. The tasks involved are the WSG server jobs, the user job (one per client), the communications router tasks, the virtual terminal task, and the Telnet task.

14.2.1 WSG Performance Recommendations

1. **WSG CPU Time and Capacity:** WSG involves a significant amount of CPU processing compared with other workstation connectivity types. Typical commercial transactions with WSG use significantly more CPU time than Telnet 5250 running the same application (on the target system). Therefore, it is recommended to use WSG casually or for non-performance critical transactions.

2. **WSG Response Time:** Response time is made up of browser time, network time, and server time. Browsers typically contribute 0.5 to 2 seconds to response time. Network time can vary widely, and AS/400 server time typically depends on model, workload, and application.

   A problem in any one of these areas can cause a significant performance problem for an end-user. To an end-user, it may seem apparent that any performance problem is attributable to the server, even though the problem may lie elsewhere.

   Better response time may be achieved by using the STYLE button on the browser (WSG window) to remove the F-keys on the bottom. This reduces the browser time to process and show the display.

3. **WSG Configuration:** Default settings for the WSG related parameters typically work fine. Changing some of the following parameters may improve performance:

   - The number of clients per WSG server job (CHGWSGA command): This parameter controls the maximum number of users that can be attached to a given WSG server job. When the WSG server is started, several server jobs start. As clients sign on, they are assigned to these server jobs (one server is assigned to its maximum before users are assigned to the next server).

     It is unlikely that this parameter needs to be changed. Increasing this parameter may be recommended if there are hundreds of casual users. Decreasing this parameter may be recommended if there are several heavy users.

- MAXFRAME (on LIND): The maximum frame size parameter can be increased from 1994 bytes for TRLAN (or other values for other protocols) to its maximum of 16393 bytes to allow for larger transmissions.
- MTU size (CFGTCP command): The maximum transmission unit size parameter for both the route and interface affect the actual size of the line flows. Changing these values from 576 bytes to a larger size (perhaps the maximum of 16388 bytes) may reduce the overall number of transmissions. This needs to be considered on both the server and the Web browser. Typically, even simple displays have several thousand bytes of HTML data.

4. **WSG Memory Requirements:** Follow the faulting threshold guidelines suggested in the Appendix A, “Guidelines for Interpreting Performance Data” on page 383 by adjusting the memory in the machine pool, the pool that the WSG servers run in, and the pool in which the user jobs run (use the WRKSYSSTS command). For the WSG server pool, the actual memory required per server (working set size) varies significantly by the workstation I/O characteristics of the workload. Based on simple tests conducted by ITSO Rochester use the following values as an initial starting point for the WSG server pool:

   ```
   750 KB + 750 KB for each WSG server
   ```

5. **Communications Lines and IOPs:** WSG/HTML sends significantly more bytes to the client workstation than Telnet does for the same application. Therefore, plan for additional communications line capacity to handle this extra load. For example, a Telnet transaction may consist of 1KB, while a WSG transaction may be 6KB for the same transaction.

6. **AS/400 Model Selection:** Seek the assistance of your IBM representative or the technical representative from your value added reseller for a capacity planning exercise.

   Server models may not be the best choice for WSG environments. The user job is labeled as an ″interactive″ job, while the other tasks involved (WSG servers, comm tasks) all are tagged ″non-interactive″. For a typical commercial transaction with WSG, the ″interactive″ portion of the transaction is about 20% of the total transaction′s CPU time. To avoid decreasing the effectiveness of a server model when using WSG, it is recommended that you keep the ″interactive″ part of your CPU utilization below the threshold where overall system performance is affected.

   In summary,

   - Do not use WSG in applications where response time is critical.
   - Limit WSG to casual use (low number of WSG transactions).
   - Use traditional (non-server) AS/400 models.

Appendix A. Guidelines for Interpreting Performance Data

Table 33. Resource Utilization Guidelines

Resource Description	Good	Acceptable	Poor
CPU 1 Processor (1)	<0.70	0.70-0.80	>0.80
CPU 2 Processors (1)	<0.76	0.76-0.83 (2)	>0.83 (2)
CPU 3 Processors (1)	<0.79	0.79-0.85 (2)	>0.85 (2)
CPU 4 Processors (1)	<0.81	0.81-0.86 (2)	>0.86 (2)
DISK ARM (9336/9337)	<0.50	0.50-0.60	>0.60
DISK IOP	<0.60	0.70	>0.80
IOP Local	<0.25	0.35	>0.40
IOP Multifunction	<0.35	0.45	>0.50
IOP Communications	<0.35	0.45	>0.50
IOP LAN	<0.35	0.40	>0.50
LINE Remote	<0.30	0.35	>0.40
FSIOP Read/Write Cache Hit (3)	>0.90	0.90	<0.90
FSIOP OS/2 CPU Utilization (3)	<0.80	0.80	>0.80

Note:

1. This refers to CPU utilization of jobs whose priorities are equal to or higher than the interactive job priorities.
2. In a multiple processor environment, the guidelines are more sensitive to utilization above the values specified in the ″good″ column.
3. You need to query file QAPMIOPD to obtain these values. See Section 13.5.8, "Integrated PC Server Performance Monitor Data Queries" on page 354 for sample queries.

Table 34. Machine Pool, Non-Database Page Faults

Main Storage Size	Good	Acceptable	Poor
All Systems	<10	10-15	>15

Table 35. Sum of Database and Non-Database Page Faults for Each Pool

Model	Good	Acceptable	Poor
9402/400 2130 2131 2132 2133	<50	50-100	>100
9406/500 2140 2141	<50	50-100	>100
9406/510 2144 9406/530 2150 50S/2121	<150	150-300	>300
9406/530 2151 2152 2153 53S/2154 2155 2156	<200	200-325	>325

Table 36. Sum of Database and Non-Database Page Faults in All Pools

Model	Good	Acceptable	Poor
9402/400 2130 2131 2132 2133	<75	75-125	>125
9406/500 2140 2141	<75	75-125	>125
9406/500 2142 2143 40S/2110 50S/2120	<150	150-350	>350
9406/510 2144 9406/530 2150 50S/2121	<300	300-600	>600
9406/530 2151 2152 2153 53S/2154 2155 2156	<400	400-650	>650

Table 37. Ratio of Wait-to-Ineligible/Active-to-Wait

Good	Acceptable	Poor
<.1	.1-.25	>.25

Table 38. Pool Size and Activity Level

Main Storage Size (MB)	Guide for Batch Job Pool (KB) and Activity Level (max no. batch jobs)	Interactive Pool Activity Level Factor (KB)
≤12	500/2-1250/3	450
16-28	1500/4-2350/5	900
32-48	2700/5-4000/6	1600
56-192	4625/5-14000/7	2500
208-272	15000/8-18600/9	3000
288-384	19500/9-23100/10	3500
416-512	24000/15-27900/15	4000
512-768	27900/15-37500/20	4500
768-1024	37500/20-47100/20	5000
1024-1280	47100/20-56000/25	5500
1280-1536	56000/25-66300/30	6000

Note: The **QINTER Activity Level Factor** can be used as a starting point to determine the activity level for the memory pool for QINTER. Divide the memory pool size by the appropriate QINTER Activity Level Factor for an initial activity level estimate.

A.1 System Capacities for AS/400 PowerPC Advanced Systems

The following tables show the system capacities for the AS/400 Advanced System models. For comparison, there are two tables, one showing the new AS/400 Advanced System using PowerPC technology and the current AS/400 Advanced System using CISC processor technology. Relative Performance Ratings are obtained using the RAMP-C workload, which is a commercial application. Some workloads may show different performance characteristics when moved to PowerPC technology. An example is application compute-intensive workloads; these are, for example, applications that implement complex business rules for decision making and are typically application compute-intensive, as are financial modelling applications that do a significant number of numeric calculations. For system capacity information on earlier models that are not covered in these tables, see Appendix A, *AS/400 Performance Management V3R1*, GG24-3723-02.

Table 39. System Capacities and Relative Performance of AS/400 Advanced System (PowerPC Technology)

	Feat Code	**Memory Capacity (MB) min - max**	**Disk Capacity (GB) min - max**	**Relative Performance Ratings**		**V3R7 CPW Value**
				Internal Processor	**System (RAMP-C)**	
Model 400	2130	32 - 160	1.96 - 23.6	3.8	4.1	13.8
	2131	32 - 224	1.96 - 23.6	5.7	6.1	20.6
	2132	32 - 224	1.96 - 23.6	8.3	8.7	27.0
	2133	32 - 224	1.96 - 23.6	10.6	10.9	33.3
Model 500	2140	64 - 768	1.96 - 151	5.7	6.4	21.4
	2141	64 - 768	1.96 - 151	8.3	9.3	30.7
	2142	64 - 1024	1.96 - 151	11.4	12.6	43.9
Model 510	2143	256 - 1024	1.96 - 318.7	19.7	21.6	77.7
	2144	256 - 1024	1.96 - 318.7	26.6	28.5	104.2
Model 530	2150	512 - 4096	1.96 - 520	32.9	37.4	131.1
	2151	512 - 4096	1.96 - 520	43.4	48.9	162.7
	2152	512 - 4096	1.96 - 520	66.6 2	74.0	278.8
	2153	512 - 4096	1.96 - 520	101.4 4	119.2	459.3
	2162	512 - 4096	1.96 - 520			509.9

Note: Relative performance is compared to an AS/400 9404 Model B10. Processor ratings that have a n symbol, show the number of CPUs contained in that model.

The type and number of disk devices, the number of workstation controllers, the amount of memory, the system model, other factors, and the application being run determine what performance is achievable.

A.2 System Capacities for AS/400 PowerPC Advanced Servers

AS/400 Advanced Server models have different performance characteristics when compared to traditional AS/400 models. Performance depends on the type of work being done. The system is optimized for server workloads in a client/server environment and for batch or non-interactive workloads. As can be seen from the following tables, batch (or non-interactive) jobs have a much higher processor rating than interactive jobs. Interactive jobs are defined as those that do screen I/O to a display, irrespective of the display being fixed function or a PC using screen emulation (for example, RUMBA).

Table 40. System Capacities and Relative Performance of AS/400 Advanced Server PowerPC Systems (RISC)

	Feat Code	Memory Capacity (MB) min - max	Disk Capacity (GB) min - max	Relative Performance Ratings Interactive/Non-interactive	
				Internal Processor	V3R7 CPW
Model 40S	2109	32 - 224	1.96 - 23.6	2.6/8.3	9.4/27.0
	2110	32 - 224	1.96 - 23.6	3.8/10.6	13.8/33.3
	2111	64 - 512	1.96 - 23.6		20.6/59.8
	2112	64 - 512	1.96 - 23.6		30.7/87.3
Model 50S	2120	64 - 1024	1.96 - 318.7	5.7/19.7	21.4/77.7
	2121	64 - 1024	1.96 - 318.7	8.3/26.6	30.7/104.2
	2122	64 - 1024	1.96 - 318.7		30.7/130.7
Model 53S	2154	512 - 4096	1.96 - 520	8.3/43.4	30.7/162.7
	2155	512 - 4096	1.96 - 520	8.3/66.6 2	30.7/278.8 2
	2156	512 - 4096	1.96 - 520	8.3/101.4 4	30.7/459.3 4
	2157	512 - 4096	1.96 - 520		30.7/509.9 4

Note: Relative performance is compared to an AS/400 9404 Model B10. Processor ratings that have an n symbol show the number of CPUs contained in that model.

The type and number of disk devices, the number of workstation controllers, the amount of memory, the system model, other factors, and the application being run determine what performance is achievable.

Table 41. Queuing Multiplier Based on CPU Utilization

u%	QM (1 Processor)	QM (2 Processors)	QM (3 Processors)	QM (4 Processors)
70	3.33	1.96	1.52	1.32
75	4.00	2.29	1.73	1.46
80	5.00	2.78	2.05	1.69
85	6.66	3.60	2.59	2.09
90	10.00	5.26	3.69	2.91
95	20.00	10.25	7.01	5.39

Note: As the Queuing Multiplier increases above four, performance problems with the resource (CPU) should be expected. This is a conservative guideline as multi-processors can deliver acceptable performance with higher Queuing Multiplier values. The PRTTNSRPT - Job Summary report shows the CPU Queuing Multiplier.

Appendix B. Program Exceptions

Systems prior to V3R6 report a number of exception types through the use of performance tools Component Report and the Advisor. The reports show the number of exceptions per second per interval that occurred. The Advisor shows the percent of the CPU used by exceptions in an interval. In addition, there are a set of charts that show the percent of the CPU used as a result of ″n″ number of exceptions by type per second.

In V3R6, some of the exceptions have been eliminated, and some of the exception CPU overhead has changed as a result of the machine implementation or the type of program in which they occur.

B.1 Program Exceptions

EAO These are gone in V3R6 as a result of the difference in PowerPC hardware addressing structure. Any data field or report that had them has been changed to indicate their absence. See Section B.4, “Removal of Effective Address Overflow (EAO) Exceptions in V3R6” on page 391 for more details on why we had EAO exceptions and why with using PowerPC technology and 64-bit addressing, they no longer occur.

Size These are the result of an arithmetic operation in which the receiving field is too small for the result. They are an application programming problem and still occur.

The programs should be reviewed and changed to ensure that the proper receiver field specification is used or that the programming algorithm is doing the function in a manner to avoid size exceptions. Using Performance Monitor trace data and Transaction Report (PRTTNSRPT RPTTYPE *TRSIT) may provide additional information about which programs are getting size exceptions.

Verify Verify exceptions occur when trying to resolve an as yet unresolved pointer.

This exception can occur on the RISC machine and is an application programming problem. The program should be changed to ensure that the variable used in a CALL instruction does not change from one use to the next. Use PEX STATS and PEX TRACE to find out where it is occurring.

Authority

These can occur on the PowerPC AS/400 systems and are the result of a system security setup mismatch.

The same rules apply to fixing these as on pre-V3R6 systems. Ensure that objects do not have private authorities on them that are less than the PUBLIC authority. For example, PUBAUT(*CHANGE) and QPGMR(*USE) causes authority checking to be done.

Decimal Data

These can occur on PowerPC AS/400 systems. This is usually related to incorrect data specification in application data migrated from other systems, especially the System/36.

B.2 CPU Exception Figures

At the time of publication, no new figures have been produced for AS/400 models with PowerPC technology.

Until these figures have been produced and normalized back to AS/400 V3R2 figures, we have produced a set of tables based on the V3R2 figures. On lab and production systems, we have found that the CPU exception figures for the same CPW rating are similar for PowerPC and non-PowerPC AS/400 models. It is quite possible that the PowerPC is more efficient and uses slightly less CPU.

The following tables are based on non-PowerPC figures and calibrated for PowerPC AS/400 models based on CPW ratings.

In OS/400 V3R7, there was an improvement made to authority lookups where the last 32 authority lookups in each job were cached. For jobs that access many objects, especially small modules in ILE, this can have a significant improvement when private authorities are being used. Currently, we have no definitive figures on the relative improvement of this new V3R7 feature.

B.3 Exception CPU Use Tables

The following sets of tables show CPU cost for size exceptions, verify exceptions, and authority exceptions. These are based on benchmarks originally using non-PowerPC processors and scaled using CPW ratings. The programs were written using OPM compilers. On PowerPC AS/400 models, there may be a significant difference between OPM and the various ILE environments **so these tables should be used as guide only.**

Table 42. Size Exceptions versus CPU % Cost - 1 of 2

Exceptions per Second	**2162**	**2153**	**2152**	**2151**	**2150**	**2144**	**2143**
100	0	0	0	0	0	1	1
500	1	1	1	2	2	2	3
1000	1	1	2	3	4	5	6
2000	2	2	3	6	7	9	12
3000	3	3	5	9	11	14	18
5000	5	5	9	15	18	23	31
10000	9	10	17	29	36	46	61
15000	14	16	26	44	55	69	92
20000	19	21	34	59	73	91	
30000	28	31	51	88			
50000	47	52	85				
70000	65	73					

Table 43. Size Exceptions versus CPU % Cost - 2 of 2							
Exceptions per Second	**2142**	**2141**	**2140**	**2133**	**2132**	**2131**	**2130**
100	1	2	2	1	2	2	4
300	3	5	7	4	5	7	10
500	5	8	11	7	9	12	17
1000	11	16	22	14	18	23	35
2000	22	31	45	29	35	46	69
3000	33	47	67	43	53	69	
5000	54	78		72	88		
7000	76						

Table 44. Verify Exceptions versus CPU % Cost - 1 of 2							
Exceptions per Second	**2162**	**2153**	**2152**	**2151**	**2150**	**2144**	**2143**
100	0	0	1	1	2	2	3
300	1	1	2	4	5	6	8
500	2	2	4	6	8	10	13
1000	4	4	7	12	15	19	26
2000	8	9	14	24	30	38	51
3000	12	13	21	37	45	57	77
5000	19	22	36	61	76	95	
7000	27	30	50	85			
10000	39	43	71				
15000	58	65					
20000	78	86					
25000	97						

Table 45. Verify Exceptions versus CPU % Cost - 2 of 2							
Exceptions per Second	**2142**	**2141**	**2140**	**2133**	**2132**	**2131**	**2130**
25	1	2	2	2	2	2	4
50	2	3	5	3	4	5	7
100	5	7	9	6	7	10	14
300	14	19	28	18	22	29	43
500	23	32	46	30	37	48	72
700	32	45	65	42	51	67	
1000	45	65	93	60	73	96	
1500	68	97		89			
2000	90						

Table 46. Authority Lookup Exceptions versus CPU % Cost - 1 of 2

Exceptions per Second	2162	2153	2152	2151	2150	2144	2143
25	0	0	0	0	0	0	1
50	0	0	0	0	1	1	1
100	0	0	1	1	1	2	2
300	1	1	2	3	3	4	6
500	2	2	3	5	6	7	9
1000	3	3	5	9	11	14	19
3000	9	10	16	27	34	42	57
5000	14	16	26	45	56	70	94
7000	20	22	37	63	78	98	
10000	29	32	53	90			
15000	43	48	79				

Table 47. Authority Lookup Exceptions versus CPU % Cost - 2 of 2

Exceptions per Second	2142	2141	2140	2133	2132	2131	2130
25	1	1	2	1	2	2	3
50	2	2	3	2	3	4	5
100	3	5	7	4	5	7	11
300	10	14	21	13	16	21	32
500	17	24	34	22	27	36	53
700	23	33	48	31	38	50	74
1000	33	48	68	44	54	71	
2000	67	95		88			
3000	99						

Table 48. Decimal Data Error Exceptions versus CPU % Cost - 1 of 2

Exceptions per Second	2162	2153	2152	2151	2150	2144	2143
250	0	0	1	1	1	1	2
500	0	1	1	2	2	3	3
1000	1	1	2	3	4	5	7
3000	3	4	6	10	12	15	21
5000	5	6	10	16	20	26	34
10000	10	12	19	33	41	51	69
20000	21	23	38	65	81		
30000	31	35	57	98			
40000	42	46	76				
50000	52	58	95				

Table 49. Decimal Data Error Exceptions versus CPU % Cost - 2 of 2

Exceptions per Second	2142	2141	2140	2133	2132	2131	2130
50	1	1	1	1	1	1	2
100	1	2	3	2	2	3	4
250	3	4	6	4	5	7	10
500	6	9	12	8	10	13	19
700	9	12	17	11	14	18	27
1000	12	17	25	16	20	26	39
2000	24	35	50	32	39	52	77
4000	49	69	99	64	79		
6000	73			96			
8000	97						

B.4 Removal of Effective Address Overflow (EAO) Exceptions in V3R6

Effective address overflow exceptions on the S/38 and CISC AS/400 systems result from two different views of the six-byte address. This address really has two parts, the segment identifier that identifies a unique group of virtual addresses (called a segment or segment group) and an offset that identifies specific locations within the segment. S/38 hardware was designed with the assumption that the system wants to use a larger number of smaller segments. Thus, the hardware treated the six-byte address as four bytes of segment identifier and two bytes of offset. This allowed over one trillion segments each with a maximum size of 64KB. The software designers felt that the system needed a smaller number of larger segments and treated the address as three bytes of segment identifier and three bytes of offset that allows over 16 million segments each with a maximum size of 16MB.

The hardware detected any operation that spanned a 64K boundary and raised an effective address overflow exception when this happened. According to the hardware′s view of the address, this was always a bad thing in that someone had tried to cross from their segment into what was potentially someone else′s segment. But due to the software design, this was generally not a problem. For example, a program could be storing a piece of data that crossed from the first 64K of a segment into the second 64K of the segment. (Note that the 16MB segment viewed by the software can contain up to 256 of the 64K ″hardware″ segments.) Thus, the software needed to look at EAO exceptions and decide if they were bad ones or not. Because IMPI operations cannot use operands spanning a 64K boundary, this handler also needed to look at the good ones and decompose the instruction into pieces that did not span a 64K boundary. For example, a Move Character (MVC) operation might have a target operand of 10 bytes that spanned a 64K boundary. This move needed to be split into two MVC operations, one to move the first part prior to the 64K boundary and one to move the second part after the 64K boundary. Needless to say, this software EAO exception handler could and did impact performance at times.

Because the AS/400 PowerPC hardware is a new design using eight-byte addresses, there is no longer a mismatch between the hardware and software views of the address. There is no longer any need for the software handler previously described and no corresponding performance impact. Hence, EAO exceptions are not reported for PowerPC AS/400 system.

Appendix C. IBM Internal Use Only Tools/Documents

This appendix discusses a range of tools useful for performance management and review but are restricted to IBM personnel and business partners. Customer access to these documents and programs are only through, and at the discretion of, authorized IBM personnel and business partners who determine when a customer can benefit from using any of these facilities.

The user is cautioned that some of these tools are to be used ″as is″. They are not supported by IBM, and as such, there is no commitment to resolve any problems that may arise with the use of these tools.

C.1 HONE Items/Internal Publications

Many performance related information items are available to IBM personnel and business partners. The following HONE items are of particular interest:

- 226NC
- 130NC
- RTA000085932

Many informal sources of information, such as COMMON presentations are also available on the AS4TOOLS disk. There is a wealth of information in these documents that may be useful to IBMers in the field.

Each year, the ITSO Rochester conducts AS/400 Forums in EMEA, U.S., and Asia/Pacific for IBM and Business Partner personnel. The presentations for these Forums, which include Performance Updates and Performance Tool Updates, are available from the ITSOROCH tools disk. A catalog of deliverables and instructions on how to obtain these presentations can be obtained by entering the following statement on a VM terminal:

```
TOOLS SENDTO WTSCPOK TOOLS2 ITSOROCH GET CATALOG PACKAGE
```

The *AS/400 Performance Capabilities Reference* is another useful source of performance information. An authorized IBM representative may obtain the latest copy of this document by entering the following statement on a VM terminal:

```
REQUEST V3R2 FROM FIELDSIT AT RCHVMW2 (your name       <----V3R2

                        or

TOOLCAT MKTTOOLS GET AS4PPCPF PACKAGE                  <----V3R7
```

C.2 AS/400 Quick Sizer

QSIZE400 is, as the name implies, a quick sizing tool of moderate complexity. It is available on HONE or IBMLink to IBMers and IBM business partners. It can also be downloaded and run on an IBM compatible personal computer.

QSIZE400 supports manually entering up to five workloads derived from measured profiles generated by the Performance Tools licensed program. It also contains about 35 predefined profiles that can be used to create a composite profile that represents the proposed workload the AS/400 system is to be subjected to. These predefined workloads include:

- System activity
- Spool activity
- Batch activity
- RAMP-C
- IBM office benchmark
- Correspondence center activity
- 5250 Display Station Pass-Through
- 3270 device emulation activity
- SQL and Query/400 activities
- Programmer activity
- PC workloads, and so on

Once the planned workload is developed using measured profiles or predefined profiles, QSIZE400 can be used to estimate the capacity of the AS/400 system that meets the objectives of the user.

Note: The measured profiles from the Performance Tools facility does not include any PC activity or target pass-through workload that may have been on the current system.

C.3 AS/400 IBMLIB

IBMLIB is a set of internally designed tools to assist in analyzing performance is some unique situations. It also has some elegant graphing support to assist System Engineers in zeroing-in on performance related problems. The tools run exclusively on an AS/400 system and an IPDS printer is required to use the graphics support. There is an OS/400 V3R6 version and an OS/400 V3R7 version.

The graphics support in IBMLIB can be used as an alternative or supplement to the graphics support in the Performance Tools licensed program 5716-PT1. But you are cautioned that IBMLIB is shipped ″as is,″ with no current plan for resolving problems.

We have included an explanation of some of the functions in the IBMLIB that we find most useful as a help to document performance data graphically, or to trace the system for authority lookups.

The full description of functions in library IBMLIB can be found in the file SOURCE, member README.

IBMLIB can be obtained electronically through a **VM PROFS REQUEST**.

```
REQUEST IBMLIBAS FROM CRAVENS AT RCHVMP (your-name
```

This sends an AS/400 SAVF of library IBMLIB to your VM USERID. If you have an AS/400 system in your VM network, you can use the VM SENDFILE command to send it to your AS/400 system.

The following commands are available with IBMLIB:

ACCGRPMON	Run DSPACCGRP monitor
AUTHPRT	Print Authority Analysis Report
AUTHTRC	Run Authority Trace via TRCINT
DISKUTIL	Query Disk Utilization
DMPTCMD	Fix up QAPMDMPT trace member command
EXCEPTION	Exception CLP call except
EXPNPRT	Print Exception Analysis Report
EXPNTRC	Run Exception and/or Authority Trace
GPHCMD	Graph command
GPHCXD	Graph command
GPHCZD	Graph command
GPHSMRY	Create a graph summary data file
GPHSXRY	Create a graph summary data file
GPHSZRY	Create a graph summary data file
JOBPRTY	Print job summary by priority
PRTPRFRPT	Print Performance Reports/Graphs
PRTTRCSUM	Print Trace Summary
QRYCMD	Query command
QRYJOBS	Print Query over QAPMJOBS file
SLTJOB	Select Jobs for BGU analysis CMD
TRSITDET	Create Transition Detail Records CMD
XFOIL	XFOIL command

C.3.1 Detailed Steps for Analyzing Data

Once performance monitor data has been collected, there are just two steps required to produce graphical output.

C.3.1.1 Step 1: Create Summary File

You must create a summary file before you can run any analysis or graphing commands in this package. Just add IBMLIB to your library list and run the GPHSMRY command with prompting. Since this can be a long running program depending on the amount of data collected, you may want to submit this command to batch.

Supplying the collection library name as the INPLIB parameter, up to 20 characters for the CUSNAM parameter and using the defaults for the rest of the parameters should be sufficient most of the time. The following example uses the GPHSMRY command:

```
GPHSMRY INPLIB(IBMR2DTA) CUSNAM('customer name')
```

See Section C.3.2, “GPHSMRY Command Details” on page 397 for a full description of this command and all of its parameters.

C.3.1.2 Step 2: Create Summary Overview Graph

Use the GPHCMD command with the default options to produce graphs that show CPU utilization by priority over time.

The reason for using the GPHCMD command with the default options first is to gain a quick perspective on how much data you actually have, when it was collected, and whether the CPU utilization is significant. You may decide to eliminate some members from further analysis by either deleting members through the performance tools facility and rerunning the GPHSMRY command in Step 1, or simply use date/time selection when running additional graphing analysis.

The following example uses the GPHCMD command:

```
GPHCMD  INPLIB(IBMR2DTA)
```

See Section C.3.3, "GPHCMD Command Details" on page 398 in this publication for a full description of this command and all of it′s parameters. Note that the GPHCMD command can also be used to create a series of more detailed analysis graphs if required.

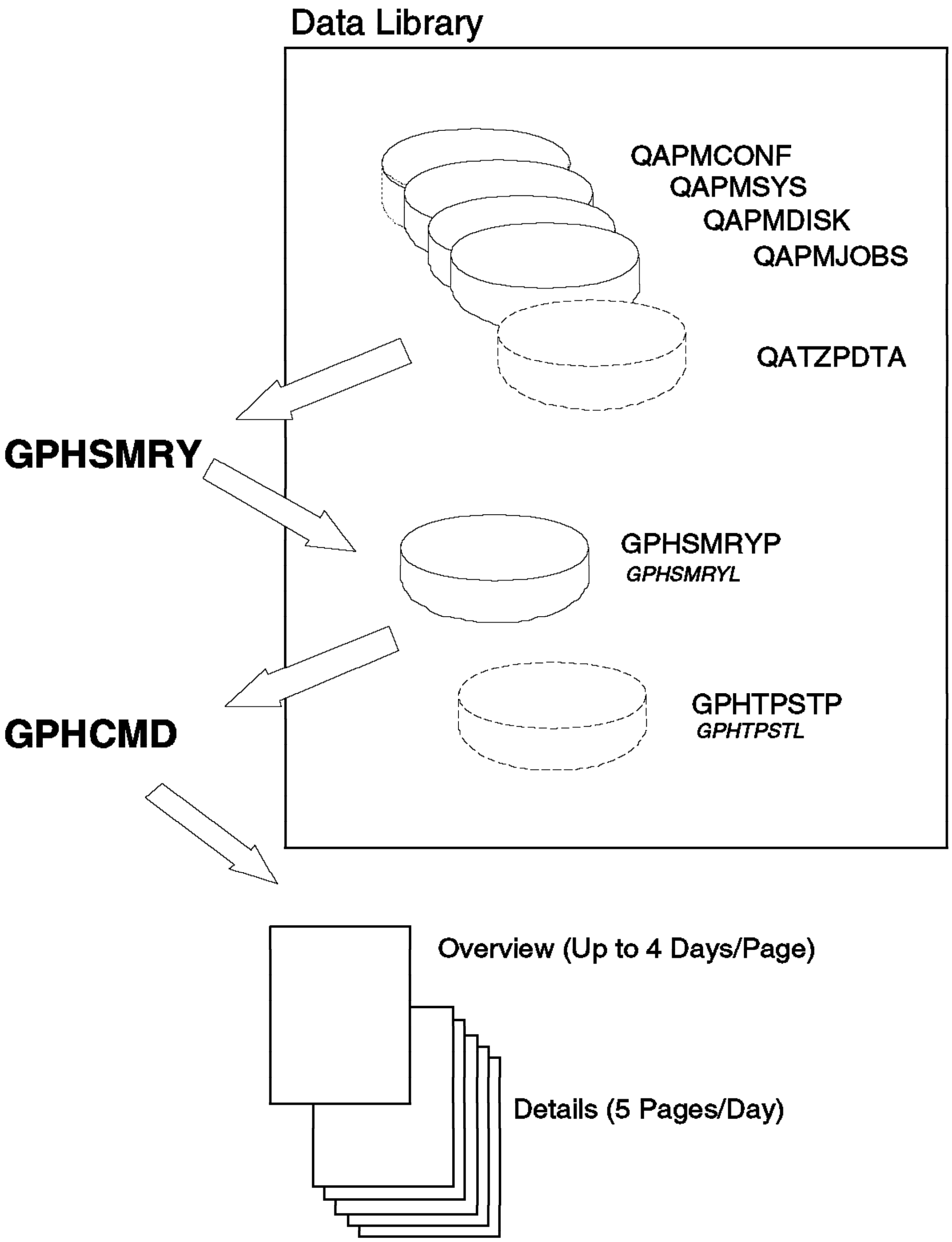


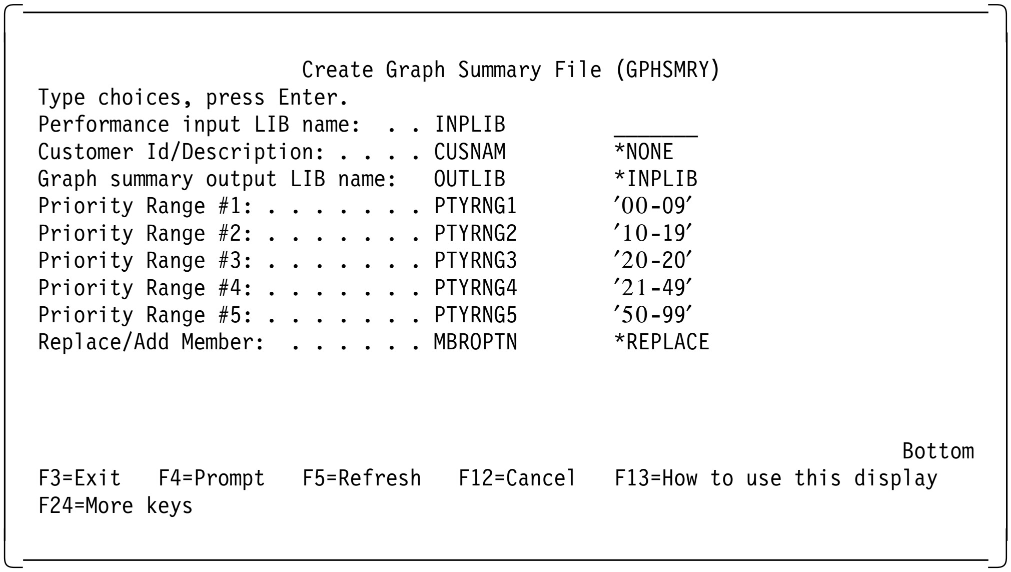
Figure 68. Schematic of Steps Required to Analyzing Data

C.3.2 GPHSMRY Command Details

The GPHSMRY command is used to create a summary file (GPHSMRYP) that is used by the GPHCMD command and PRTCMD command. Data is summarized from all the members in the QAPMSYS, QAPMCONF, QAPMJOBS, and QAPMDISK performance files in the named collection library.

Note that the default is to replace the contents of the summary file each time the command is executed. To create a summary file, use the following instructions:

- Add IBMLIB to your library list (ADDLIBLE IBMLIB).
- Run the GPHSMRY command with prompting.



```
                       Create Graph Summary File (GPHSMRY)
Type choices, press Enter.
Performance input LIB name:  . . INPLIB          _______
Customer Id/Description: . . . . CUSNAM          *NONE
Graph summary output LIB name:   OUTLIB          *INPLIB
Priority Range #1: . . . . . . . PTYRNG1         '00-09'
Priority Range #2: . . . . . . . PTYRNG2         '10-19'
Priority Range #3: . . . . . . . PTYRNG3         '20-20'
Priority Range #4: . . . . . . . PTYRNG4         '21-49'
Priority Range #5: . . . . . . . PTYRNG5         '50-99'
Replace/Add Member:  . . . . . . MBROPTN         *REPLACE




                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

The GPHSMRY command can be a long running program, depending on the amount of data collected. You may want to submit this command to batch.

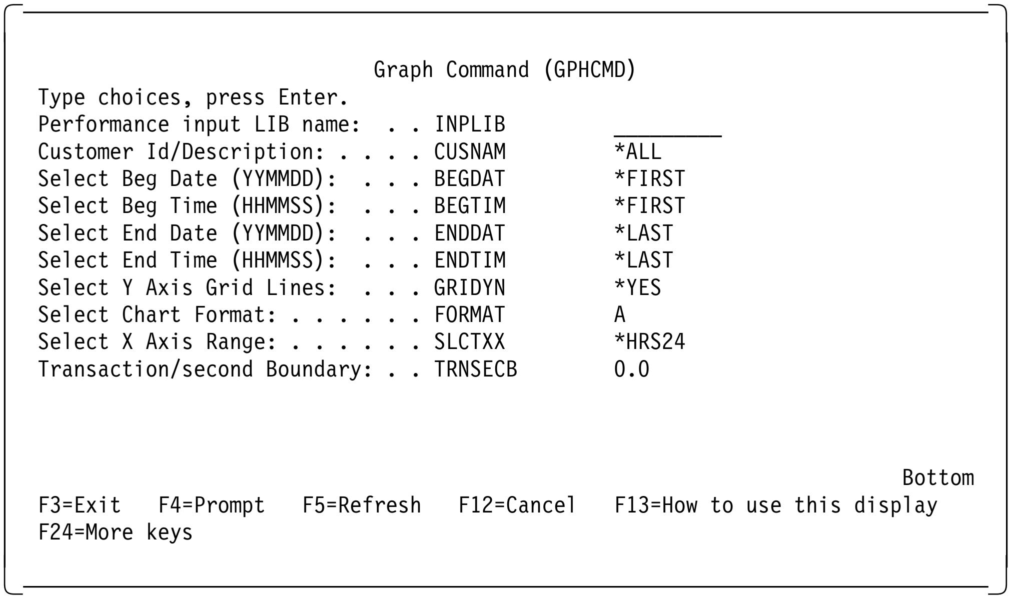
Definition of GPHSMRY Parameters:

- **INPLIB:** The INPLIB parameter allows you to specify the name of the library containing performance data that you want to graph.
- **CUSNAM:** You can specify the customer name, up to 20 characters for this parameter.
- **OUTLIB:** The OUTLIB parameter is meant to store the summary graph file in a user-defined library. The default is to store the summary graph file in the same file as the input library.
- **PTYRNG1 - PTYRNG5:** The PTYRNG1 parameter and PTYRNG5 parameter represent the lower and upper boundaries for five categories of CPU priority. The middle category (PTYRNG3) should be set with the upper and lower boundary for interactive work. You might want to change this value if the customer does not use the IBM shipped default of 20 for interactive work.
- **MBROPTN:** The MBROPTN parameter default *REPLACE allows for a new summary graph file member to be created or replaced. The *ADD option adds new records to existing records in a member. If you use *ADD, you must use the customer description to select the records your want. Stick with the default.

C.3.3 GPHCMD Command Details

The GPHCMD command is used to produce either summary or detailed graphs. To use this command, use the following procedure:

- Use the GPHCMD command and use **F4** to prompt.
- Input the parameters that are described to select the appropriate options.



```
                           Graph Command (GPHCMD)
Type choices, press Enter.
Performance input LIB name:  . . INPLIB          _________
Customer Id/Description: . . . . CUSNAM          *ALL
Select Beg Date (YYMMDD):  . . . BEGDAT          *FIRST
Select Beg Time (HHMMSS):  . . . BEGTIM          *FIRST
Select End Date (YYMMDD):  . . . ENDDAT          *LAST
Select End Time (HHMMSS):  . . . ENDTIM          *LAST
Select Y Axis Grid Lines:  . . . GRIDYN          *YES
Select Chart Format: . . . . . . FORMAT          A
Select X Axis Range: . . . . . . SLCTXX          *HRS24
Transaction/second Boundary: . . TRNSECB         0.0




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Definition of GPHCMD Command Parameters

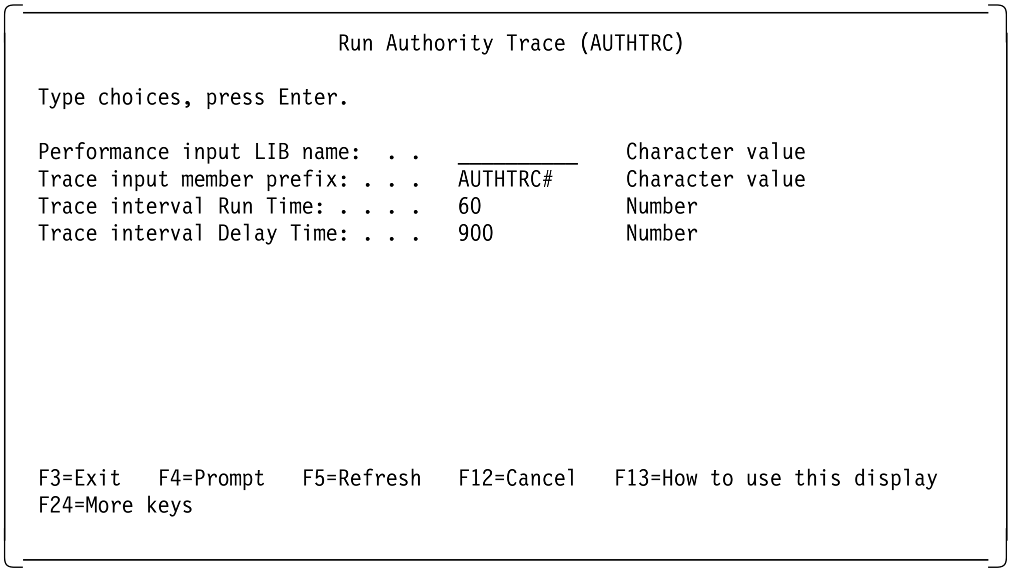
- **BEGDAT, BEGTIM, ENDDAT, ENDTIM:** These parameters allow data and time selection of data.
- **GRIDYN:** The GRIDYN parameter allows for grid lines on the Y axis.
- **FORMAT:** The FORMAT parameter has only two choices: ″A″ allows for CPU analysis over time and puts up to four days on one graph page. ″B″ causes a series of five detailed graph pages to be produced for each day, each page with up to four graphs per page.
- **SLCTXX:** This parameter is used to alter the X axis which is time. The default is 24 hours. To change this, use *SELECT and put the time values you want into the BEGTIM and ENDTIM parameters.
- **TRNSECB:** This parameter is used as a transaction filter. A measurement interval must have at least the transaction rate specified (in transactions per second) in order to be included in response time analysis.

C.3.4 AUTHTRC Command Details

Appendix A, “Guidelines for Interpreting Performance Data” on page 383 includes a table showing approximate CPU impact of authority lookups. The Performance Tools Component Report lists Authority Lookups exceptions per second. Use this table to determine if the number of Authority Lookups per second is impacting the CPU impact and is worth further investigation.

Note: There is no system or Performance Tools function to identify the objects with excessive authority lookups. The AUTHTRC command and AUTHPRT command in IBMLIB can help identify the objects with excessive private authority counts.

The AUTHTRC command utilizes the TRCINT and DMPTRC commands to start and dump an authority management trace for a user-specified measurement period. The command entry looks similar to:



```
                          Run Authority Trace (AUTHTRC)

 Type choices, press Enter.

 Performance input LIB name:  . .   __________    Character value
 Trace input member prefix: . . .   AUTHTRC#      Character value
 Trace interval Run Time: . . . .   60            Number
 Trace interval Delay Time: . . .   900           Number









 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Use the following parameter definitions to select the correct parameters. Usually the defaults can be used.

The INPLIB parameter names the library to store the trace data file QAPMDMPT that is automatically created if not present. You need to create the library before running this command if it does not exist already. You should choose a library that does not contain regular performance data files because QAPMDMPT is the same file that is used to hold regular performance tool trace data.

The INPMBR parameter is the prefix for the authority trace members that are created. The prefix is appended automatically by a two-digit number that reflects the trace interval iteration.

The RUNTM parameter specifies the authority trace measurement interval duration in seconds.

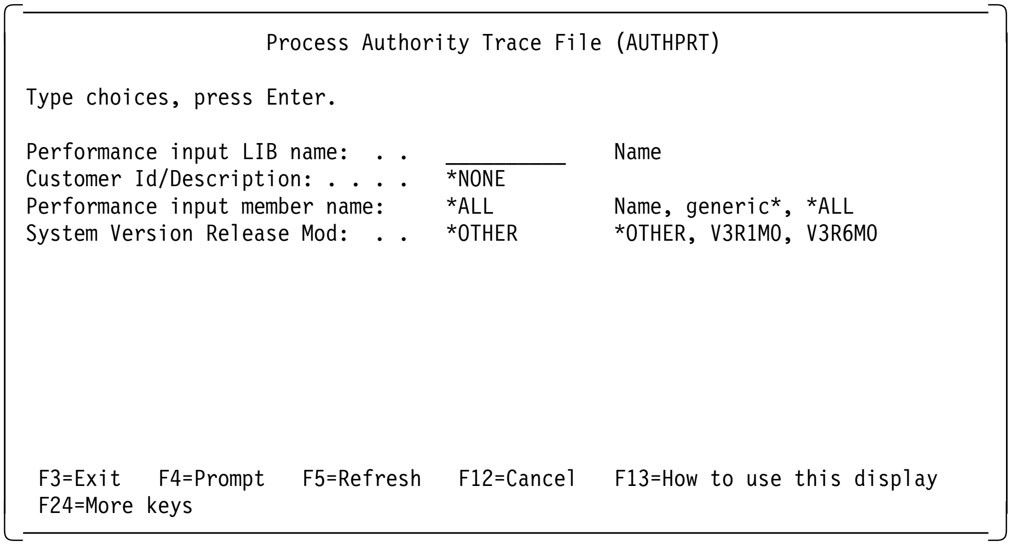
The DLYTM parameter specifies the amount of time to wait or delay between trace measurement intervals if more than one interval is chosen.

The NUMTI parameter specifies the number of trace measurement intervals to run.

Be advised that the authority management trace cannot be run when the Performance Tools monitor is running with trace turned on. This is because the authority management trace uses the same trace table as does STRPFRMON with TRACE(*ALL). On the other hand, you can run AUTHTRC at the same time as STRPFRMON if TRACE(*NONE).

C.3.5 AUTHPRT Command Details

The AUTHPRT command prints a summary report showing private authority counts by object type and by job. The command entry looks similar to:



```
                      Process Authority Trace File (AUTHPRT)

 Type choices, press Enter.

 Performance input LIB name:  . .   __________    Name
 Customer Id/Description: . . . .   *NONE
 Performance input member name:     *ALL          Name, generic*, *ALL
 System Version Release Mod:  . .   *OTHER        *OTHER, V3R1M0, V3R6M0









  F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
  F24=More keys
```

Use the following parameter definitions to select the correct parameters.

The INPLIB parameter supplies the library name where the authority trace data was placed.

The CUSNAM parameter provides a way to label the report with some user-described information such as the customer name.

The INPMBR parameter allows the selection of a specific member, a group of members, or all members in the QAPMDMPT file for the INPLIB library.

The AUTHPRT command can be run during the collection of authority trace data. The members collected up to the last trace interval are included in the report.

C.3.6 AUTHPRT Sample Output

The listings in Figure 69 and Figure 70 show private authority lookups by object and by job, and ranks them in order of frequency of authority lookup, highest to lowest ranking.

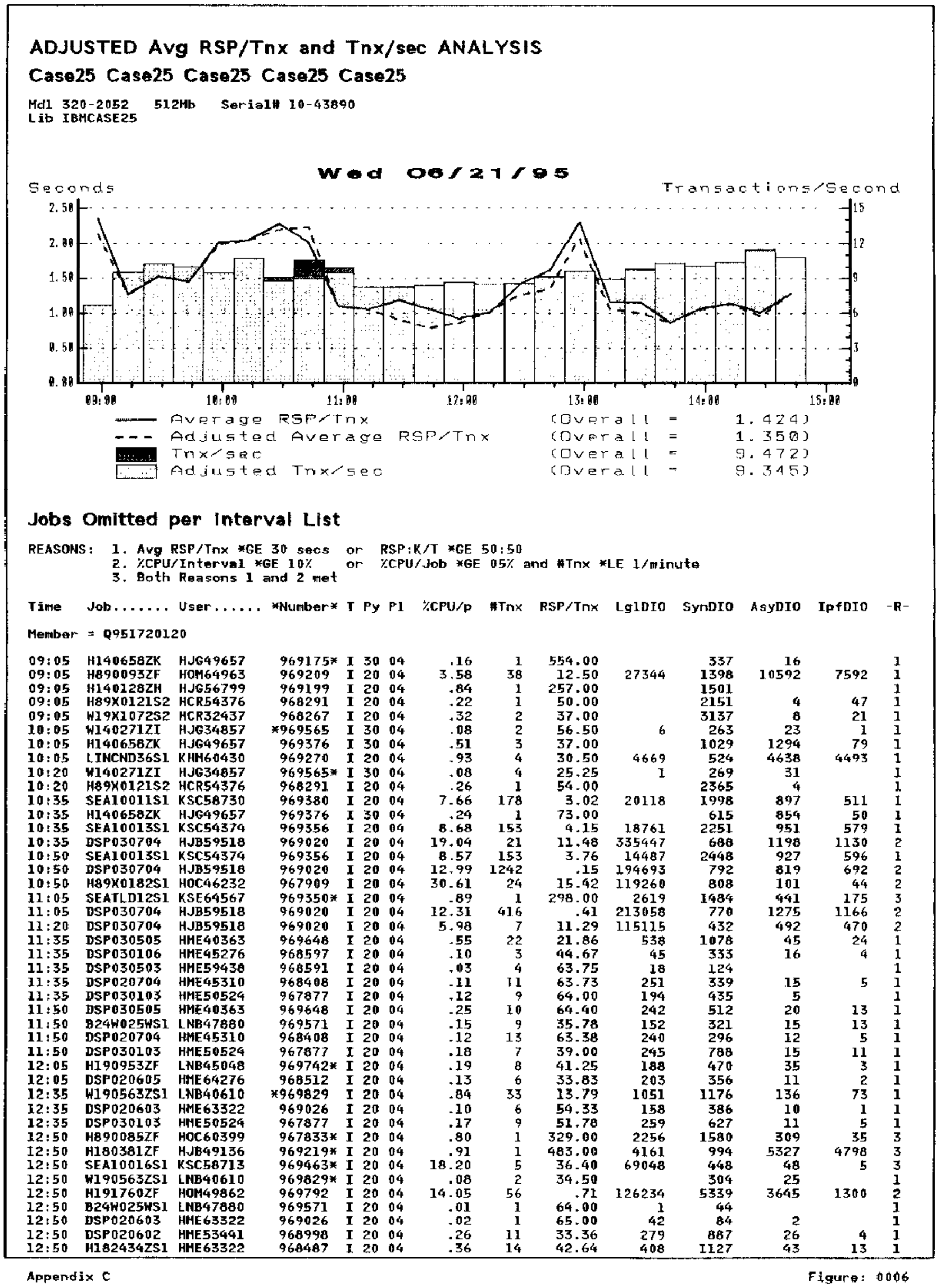
```
AUTHORITY ANALYSIS     LIBRARY: AUTHTRC     FILE: QAPMDMPT    MEMBER: AUTHTRC#01      END:  8/22/96 15:17:30.078     PAGE    1
                                                                                      BEG:  8/22/96 15:16:29.698
JCOOK BATCH TEST                                                          ELAPSED SECONDS:                60.380
OBJECT SUMMARY          AUTHORITY COUNT PER SECOND:      .513
           PRIVATE
          AUTHORITY
            COUNT     %       OBJECT       TYPE   DESCRIPTION
            -----   -----     ----------   ----   --------------------
                9   29.03     QUSER        0801   USER PROFILE
                9   29.03     QPGMR        0801   USER PROFILE
                4   12.90     QUSRSYS      0401   LIBRARY
                4   12.90     QSYS         0401   LIBRARY
                3    9.67     QGPL         0401   LIBRARY
                1    3.22     QCMD         0201   PROGRAM
                1    3.22     MAATTA       0801   USER PROFILE
 TOTAL         31
```

Figure 69. Authority Trace Example Report - Page 1 of 2

```
AUTHORITY ANALYSIS     LIBRARY: AUTHTRC     FILE: QAPMDMPT    MEMBER: AUTHTRC#01      END:  8/22/96 15:17:30.078     PAGE    2
                                                                                      BEG:  8/22/96 15:16:29.698
JCOOK BATCH TEST                                                          ELAPSED SECONDS:                60.380
JOB SUMMARY             AUTHORITY COUNT PER SECOND:      .513
           PRIVATE
          AUTHORITY
            COUNT     %       JOB        USER       NBR
            -----   -----     ---------- ---------- ------
                7   22.58     CHAINUPCW  QPGMR      055410
                6   19.35     QZDAINITP2 QUSER      055413
                6   19.35     QZDAINITP2 QUSER      055412
                6   19.35     QZDAINITP2 QUSER      055411
                4   12.90     QBRMNET    QPGMR      054890
                1    3.22     QZDAINITP2 QUSER      055408
                1    3.22     CHAINBCH   QSYS       055406
 TOTAL         31
```

Figure 70. Authority Trace Example Report - Page 2 of 2

C.3.7 Adjusted Average Response Time per Transaction Analysis



```
ADJUSTED Avg RSP/Tnx and Tnx/sec ANALYSIS
Case25 Case25 Case25 Case25 Case25

Mdl 320-2052   512Mb   Serial# 10-43890
Lib IBMCASE25

                    Wed 06/21/95

──── Average RSP/Tnx                 (Overall  =   1.424)
---  Adjusted Average RSP/Tnx        (Overall  =   1.350)
▨    Tnx/sec                         (Overall  =   9.472)
□    Adjusted Tnx/sec                (Overall  =   9.345)

Jobs Omitted per Interval List

REASONS:  1. Avg RSP/Tnx *GE 30 secs   or   RSP:K/T *GE 50:50
          2. %CPU/Interval *GE 10%     or   %CPU/Job *GE 05% and #Tnx *LE 1/minute
          3. Both Reasons 1 and 2 met

Time   Job....... User...... *Number* T Py Pl  %CPU/p  #Tnx  RSP/Tnx  LglDIO  SynDIO  AsyDIO  IpfDIO  -R-

Member = Q951720120

09:05  H140658ZK  HJG49657   969175* I 30 04     .16     1   554.00             337      16            1
09:05  H890093ZF  HOM64963   969209  I 20 04    3.58    38    12.50   27344    1398   10592    7592    1
09:05  H140128ZH  HJG56799   969199  I 20 04     .84     1   257.00            1501                    1
09:05  H89X0121S2 HCR54376   968291  I 20 04     .22     1    50.00            2151       4      47    1
09:05  W19X1072S2 HCR32437   968267  I 20 04     .32     2    37.00            3137       8      21    1
10:05  W140271ZI  HJG34857  *969565  I 30 04     .08     2    56.50       6     263      23       1    1
10:05  H140658ZK  HJG49657   969376  I 30 04     .51     3    37.00            1029    1294      79    1
10:05  LINCND36S1 KHM60430   969270  I 20 04     .93     4    30.50    4669     524    4638    4493    1
10:20  W140271ZI  HJG34857   969565* I 30 04     .08     4    25.25       I     269      31            1
10:20  H89X0121S2 HCR54376   968291  I 20 04     .26     1    54.00            2365       4            1
10:35  SEA10011S1 KSC58730   969380  I 20 04    7.66   178     3.02   20118    1998     897     511    1
10:35  H140658ZK  HJG49657   969376  I 30 04     .24     1    73.00             615     854      50    1
10:35  SEA10013S1 KSC54374   969356  I 20 04    8.68   153     4.15   18761    2251     951     579    1
10:35  DSP030704  HJB59518   969020  I 20 04   19.04    21    11.48  335447     688    1198    1130    2
10:50  SEA10013S1 KSC54374   969356  I 20 04    8.57   153     3.76   14487    2448     927     596    1
10:50  DSP030704  HJB59518   969020  I 20 04   12.99  1242      .15  194693     792     819     692    2
10:50  H89X0182S1 HOC46232   967909  I 20 04   30.61    24    15.42  119260     808     101      44    2
11:05  SEATLD12S1 KSE64567   969350* I 20 04     .89     1   298.00    2619    1484     441     175    3
11:05  DSP030704  HJB59518   969020  I 20 04   12.31   416      .41  213058     770    1275    1166    2
11:20  DSP030704  HJB59518   969020  I 20 04    5.98     7    11.29  115115     432     492     470    2
11:35  DSP030505  HME40363   969648  I 20 04     .55    22    21.86     538    1078      45      24    1
11:35  DSP030106  HME45276   968597  I 20 04     .10     3    44.67      45     333      16       4    1
11:35  DSP030503  HME59438   968591  I 20 04     .03     4    63.75      18     124                    1
11:35  DSP020704  HME45310   968408  I 20 04     .11    11    63.73     251     339      15       5    1
11:35  DSP030103  HME50524   967877  I 20 04     .12     9    64.00     194     435       5            1
11:50  DSP030505  HME40363   969648  I 20 04     .25    10    64.40     242     512      20      13    1
11:50  B24W025WS1 LNB47880   969571  I 20 04     .15     9    35.78     152     321      15      13    1
11:50  DSP020704  HME45310   968408  I 20 04     .12    13    63.38     240     296      12       5    1
11:50  DSP030103  HME50524   967877  I 20 04     .18     7    39.00     245     788      15      11    1
12:05  HI90953ZF  LNB45048   969742* I 20 04     .19     8    41.25     188     470      35       3    1
12:05  DSP020605  HME64276   968512  I 20 04     .13     6    33.83     203     356      11       2    1
12:35  W190563ZS1 LNB40610  *969829  I 20 04     .84    33    13.79    1051    1176     136      73    1
12:35  DSP020603  HME63322   969026  I 20 04     .10     6    54.33     158     386      10       1    1
12:35  DSP030103  HME50524   967877  I 20 04     .17     9    51.78     259     627      11       5    1
12:50  H890085ZF  HOC60399   967833* I 20 04     .80     1   329.00    2256    1580     309      35    3
12:50  H180381ZF  HJB49136   969219* I 20 04     .91     1   483.00    4161     994    5327    4798    3
12:50  SEA10016S1 KSC58713   969463* I 20 04   18.20     5    36.40   69048     448      48       5    3
12:50  W190563ZS1 LNB40610   969829* I 20 04     .08     2    34.50             304      25            1
12:50  H191760ZF  HOM49862   969792  I 20 04   14.05    56      .71  126234    5339    3645    1300    2
12:50  B24W025WS1 LNB47880   969571  I 20 04     .01     1    64.00       1      44                    1
12:50  DSP020603  HME63322   969026  I 20 04     .02     1    65.00      42      84       2            1
12:50  DSP020602  HME53441   968998  I 20 04     .26    11    33.36     279     887      26       4    1
12:50  H182434ZS1 HME63322   968487  I 20 04     .36    14    42.64     408    1127      43      13    1

Appendix C                                                                              Figure: 0006
```

Figure 71. Sample Output from Adjusted Average Response Time Graph

The ADJUSTED Avg RSP/Tnx and Tnx/sec ANALYSIS graph in Figure 71 shows the original and adjusted values of average response time per transaction and throughput (Transactions per second).

The objective of this graph is to identify odd and outrageous situations and adjust the transaction analysis by trimming or eliminating the contribution of those transactions. It is possible that the users and situations that surface as a result of this analysis may need to be investigated in even more detail, or controlled if possible, or simply put into separate measurement categories in order for the analyst to get a clearer picture of the more stable production workload and workload characteristics. Using this function should tend to smooth the average response time curve by eliminating and explaining the possible causes for some response time spikes.

Rule parameters have been added to the GPHSXRY command to allow an exception file to be created that contains job interval records for all interactive jobs that meet or exceed certain rule criteria. This exception file is used to adjust the overall average response time and transaction per second rate on an interval-by-interval basis.

A GPHCXD command option allows printing an ADJUSTED Avg RSP/Tnx and Tnx/sec ANALYSIS graph that shows and compares the original and adjusted values over the selected measurement time intervals. The jobs that are omitted from an interval are listed below the graph.

Jobs Omitted per Interval List

The Jobs Omitted per Interval List shows those interactive jobs that are omitted from an interval. A reason code is provided that identifies the rule used to trigger the omission. The rules are designed to select interactive jobs within intervals that exhibit the following:

1. Outrageous average response time
2. Excessive CPU utilization
3. Both of the preceding

The criteria is set as follows:

- **Rule 1:**

  This is a two-part test performed on the response time statistics for each interval of every interactive job type.

  1. **Avg Rsp/Tnx *GE 30 secs:**

     The first test is simply whether or not the average response time for all the transactions in the interval exceeds a given value. The default value on the GPHSXRY command is 30 seconds.

     For instance, job H140658ZK/HJG49657/969175 having a 554 second average response time per transaction during the first interval is omitted from the adjusted graph.

  2. **RSP:K/T *GE 50:50:**

     The second test deals with the ratio of response time (RSP) to key-think time (K/T). If the total RSP seconds for an interval equals or exceeds 50% (the GPHSXRY command default value) of the total K/T time (interval elapsed seconds -(minus) total RSP seconds), the job′s interval record is selected for omission. This test is designed to identify an interval where there was at least one transaction that had a high response time but where there were enough regular transactions to keep the average response time within the boundary of the first test.

     Consider the following three examples where each different arrival pattern gives an average response time of one second and an average key-think time of nine seconds (these averages suggest a busy user with an RSP:K/T ratio of 10:90). Let X represent one second of response time, x represent 0.5 seconds of response time, and = represent one second of key-think time or delay time between transactions.

- Example 1, uniform arrivals:

  ```
  X=========X=========X=========X=========X=========X=========
  ```

- Example 2, bursts of work between long delays:

  ```
  X==X==X===X===================================X==X==========
  ```

- Example 3, a mix of long and short response times and delays:

  ```
  X==X==XXX====================================X=====X=====X===
  ```

Now consider example 4 and example 5, which show two extremes where the RSP:K/T ratio is 50:50 and the average response time does not reflect the exceptional conditions.

- Example 4: At least one transaction with long response time occurs during the interval (this transaction could have started in this interval, in the previous interval, or spanned several intervals (or could even have been in progress before the monitor measurement started) because the RSP seconds and transaction count for a job are only updated when a transaction ends):

  ```
  XXXXXXXXXXXXXXXXXXXX====================XXX===X=X=X=X=X=X==
  ```

- Example 5: A user is either scrolling through a file, browsing a spooled report, or resting an elbow on the keyboard (here the anomaly affects the average response time (keeps it low) and the transaction rate (keeps it high)):

  ```
  X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=X=
  ```

For instance, job SEA10011S1/KSC58730/969380 with 178 transactions in the interval between 10:36am to 10:50am has an average response time of 3.02 sec per transaction.

```
Total Response time in 15 minute interval:  178 * 3.02 sec = 537.56 sec
Total Key-think time in same interval    :  900 - 537.56 sec = 362.44 sec
```

Both the maximum average response time value and the RSP:K/T ratio for Rule 1 can be adjusted using the RULE01 parameter on the GPHSXRY command. The default setting is 30,50 (30 seconds for the maximum average response time value and 50 for the %RSP portion of the RSP:K/T ratio). Making either or both of these values a high value (99) effectively turns off one or both of the tests.

- **Rule 2**

  This is a two-part test that addresses CPU utilization for each interval of every interactive job.

  1. **%CPU/Interval *GE 10%:**

     This test checks whether the job uses 10% of more (the default value) of a processor′s CPU for the measurement interval.

     For instance, job DSP030704/HJB59518/969020 has utilized 19.04% of CPU resource during the interval from 10:21am to 10:35am.

  2. **%CPU/JOB *GE 05% and #Tnx *LE 1/minute:**

     This test checks whether the job uses 5% or more (the default value) of a processor′s CPU over the jobs elapsed time during the interval (a job could have been initiated or terminated or both during a given measurement interval). In addition, the job must have run 60 seconds or longer and performed less than one transaction per 60 seconds of job elapsed run time per interval.

For instance, job DSP030704/HJB59518/969020 used 5.98% of CPU in only seven transactions during the 15 minute interval between 11:06 am to 11:20am.

Both values can be adjusted using the RULE02 parameter on the GPHSXRY command. The default setting is 10,05 (representing 10% for the first maximum job CPU utilization test and 5% for the second job CPU utilization and minimum transaction count test). Making either or both of these values a high value (99) effectively turns off one or both of the tests.

The Jobs Omitted per Interval List shows those jobs that were omitted because they met one or more of the omission criteria.

Note: This list includes an asterisk (*) next to the job number that was only present for part of the interval. The ″*″ before the job number indicated jobs initiated within that interval. The ″*″ after the job number indicated jobs that terminated within that interval. This is a new feature of the IBMLIB, and this information is not found in any of the current Performance Tools/400 reports over sample data.

C.3.8 CPU Util, Disk I/O by Priority

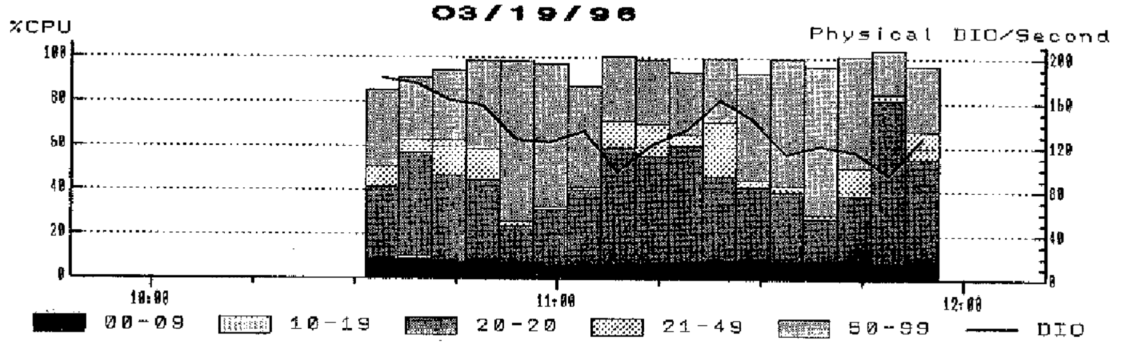


Figure 72. Sample Output from CPU Utilization and Disk I/O by Priority

The graph example in Figure 72 is for a single set of data and shows CPU utilization and Physical Disk I/Os according to classes of job priorities (shading).

C.3.9 Multiple Resources, Excluded Transaction #1

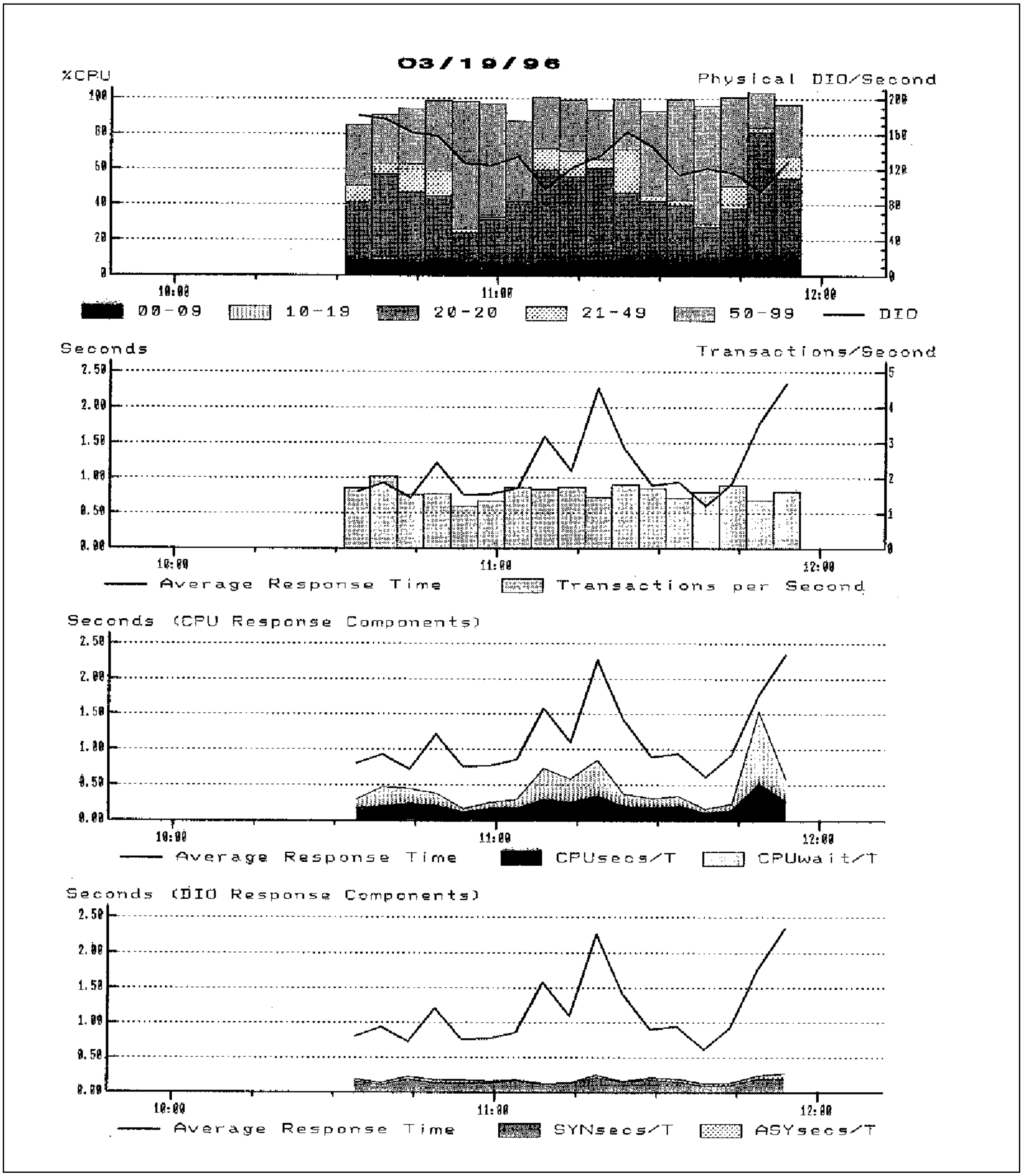


Figure 73. Sample Output from Multiple Resources, Excluded Transaction #1

The graph example in Figure 73 is for a single set of data and shows four sets of performance metric comparisons:

- CPU utilization and physical disk I/Os
- Average response time and interactive transactions per second
- Average response time, CPU seconds per transaction, and CPU wait time per transaction

- Average response time, synchronous disk I/Os per second, and asynchronous disk I/Os per second.

C.3.10 Multiple Resources, Excluded Transaction #2

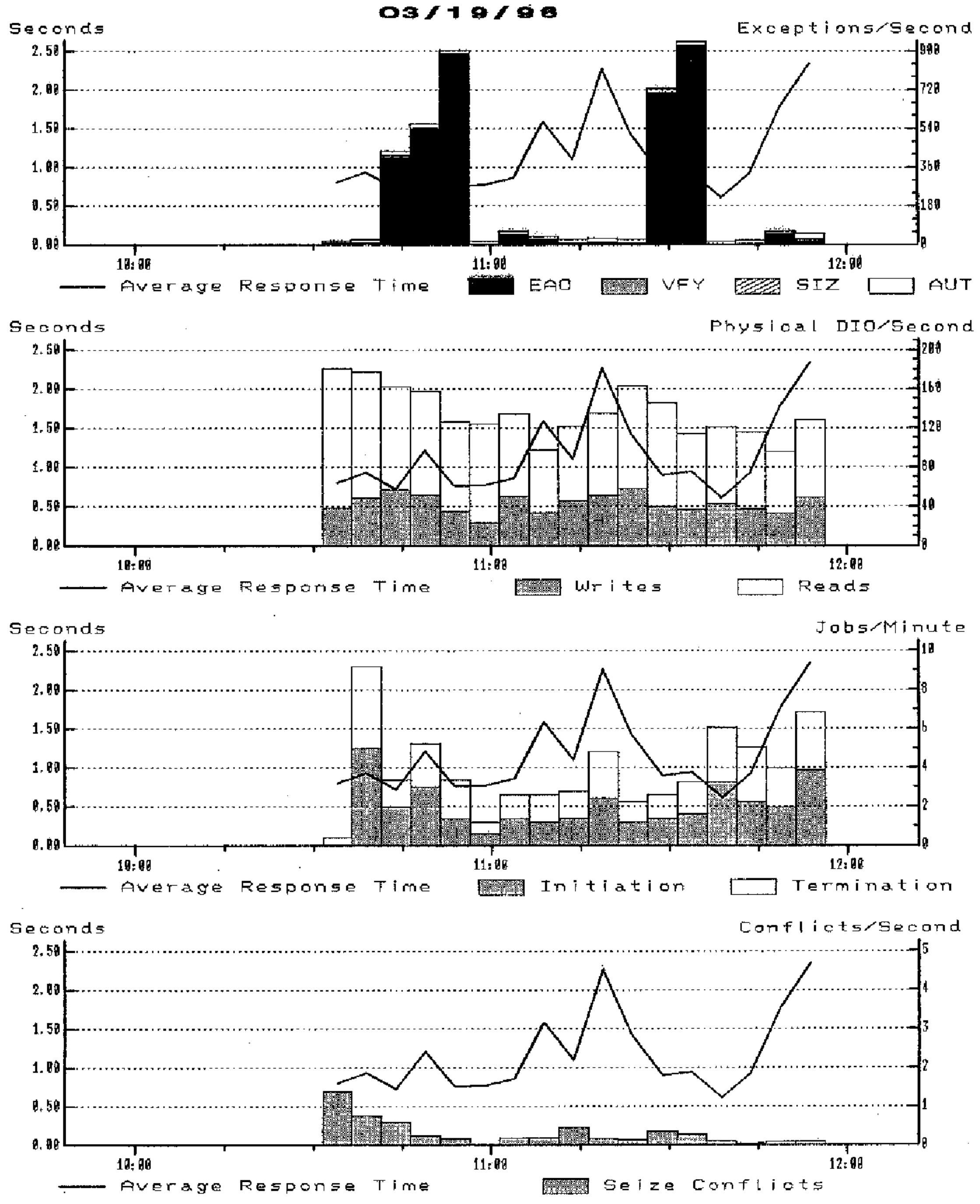


Figure 74. Sample Output from Multiple Resources, Excluded Transaction #2

The graph example in Figure 74 is for a single set of data and shows four more sets of performance metric comparisons:

- Average response time and exceptions per second
- Average response time and physical disk I/Os per second
- Average response time, job initiation, and termination
- Average response time, seize/lock conflicts per second

C.3.11 Response Time Components

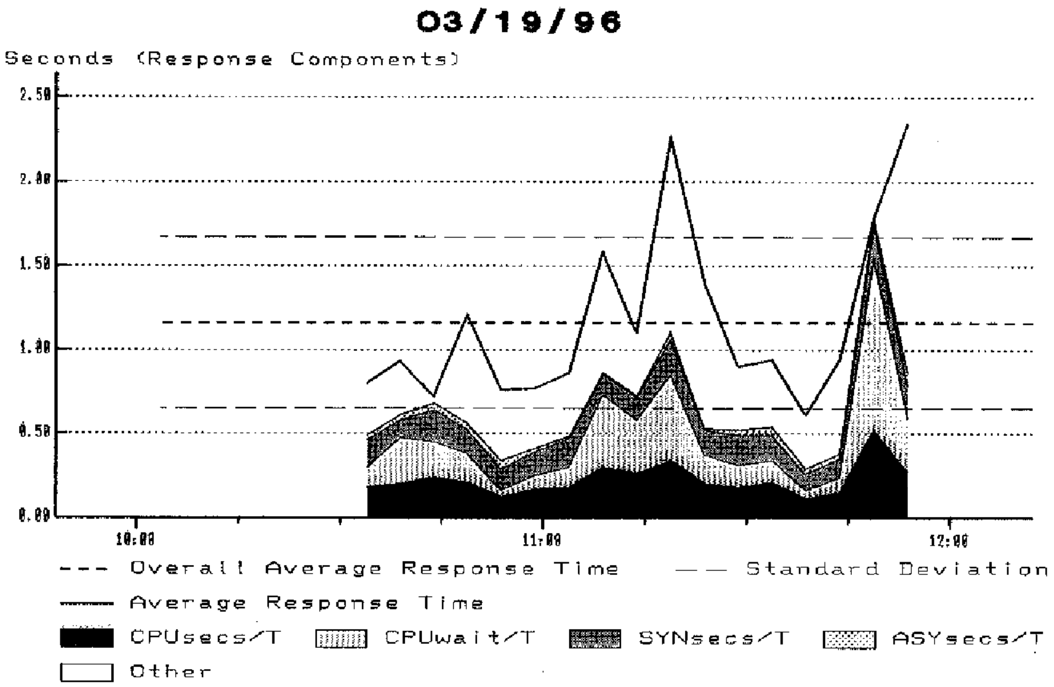


Figure 75. Sample Output for Response Time Components

The graph example in Figure 75 is for a single set of data and shows the following components of response time:

- Overall average response time
- Average response time
- Average response time standard deviation
- Average CPU seconds per transaction
- Average CPU wait time per transaction
- Average Synchronous disk I/O seconds per transaction
- Average Asynchronous disk I/O seconds per transaction
- ″Other″ or undetermined component of response time

C.3.12 Response Time, CPU Utilization by Priority

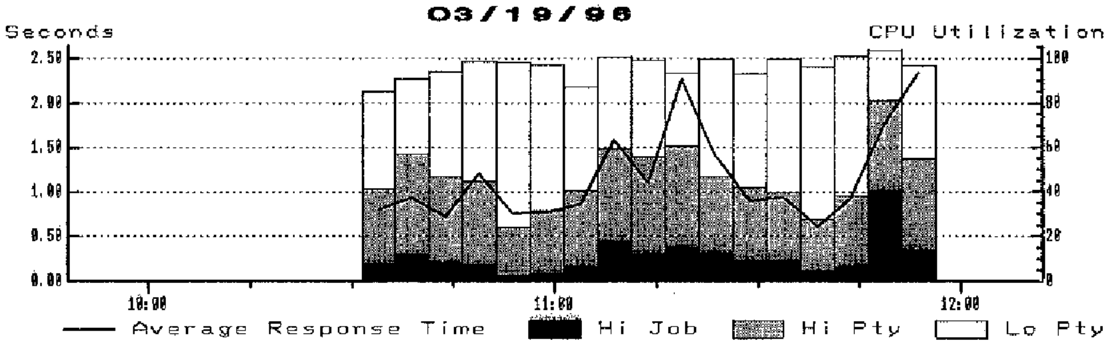


Figure 76. Sample Output for Response Time, CPU Utilization by Priority

The graph example in Figure 76 is for a single set of data and shows average response time and job CPU utilization consumed by high and low priority jobs.

C.3.13 Response Time and Logical Disk I/Os

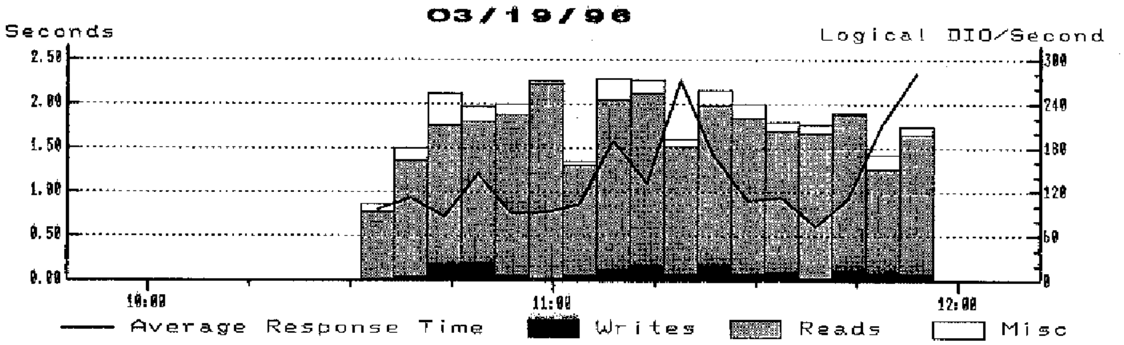


Figure 77. Sample Output for Response Time and Logical Disk I/O's

The graph example in Figure 77 is for a single set of data and shows average response time and logical disk I/Os per second for read, write, and miscellaneous I/Os.

C.4 Batch400 Overview

Batch400 is a internal tool for ″batch window (job run time) analysis″. It is supported for all releases after V2R3 including V3R6. It uses data collected by the OS/400 Performance Monitor.

C.4.1 General Description and Output

Batch400 creates a ″model file″ from the performance data and places the model into file QSBSSCHED in the named user library. The tool is asked to analyze the model and provide results that include:

- Individual batch job run time
- Run time for a group of jobs and multiple job threads
- Graphs showing threads over time of day that can be used to consider breaking up threads or reschedule the jobs
- Bar chart of thread elapsed times that can identify threads that should be reviewed ahead of other threads
- Interactive work
- Related LIC tasks disk I/Os

Job grouping and threading and job scheduling can be changed and re-analyzed.

C.4.2 How to Receive Batch400

You can obtain Batch400 by entering the following on a VM terminal session:

```
REQUEST BATCH400 FROM DANFRIZ AT RCHVMW2
```

Appendix D. Performance Tools/400 Transaction Boundary Overview

When does a transaction begin and end? Are there specific ″states″ that a transaction passes through? This appendix addresses these questions and more.

D.1 Workstation Transaction Boundaries

TheAS/400 Performance Tools/400 Guide contains information on transaction boundaries. This is a summary of this information.

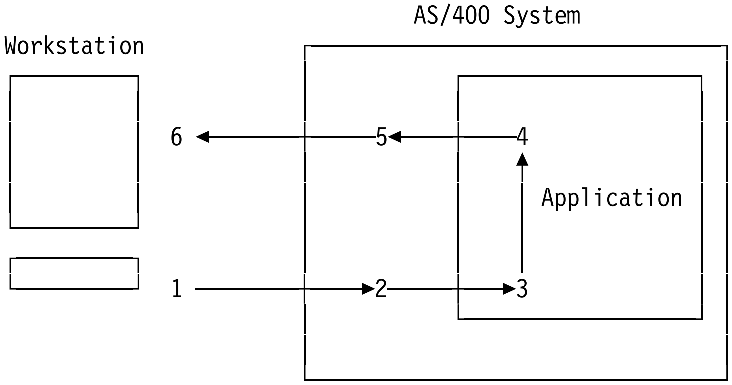


Figure 78. Interactive Transaction Boundary Flow

The numbers in the following discussion refer to Figure 78 and represent the steps a display I/O transaction goes through.

1 User presses Enter or Function key. This begins the response time period perceived by the user. However, the system does not recognize the beginning of the transaction until Step 2.

Delays are typical on a remote communication line and are dependent on the amount of current data traffic to and from other workstations on the line and how frequently the system polls the control unit for input data.

2 (SOT) Start of Transaction:

This identifies the beginning of the System Measured Response Time. Workstation I/O Management (WSIOM) processes input from the display device. This also represents the commencement of application input queuing time.

3 (SOR) Start of Resource Utilization Time:

The application must issue an input operation or accept input operation. An application program receives the data from WSIOM and begins using system resources to process the transaction. The application input queuing time ends at this point. Normally, application queuing time, such as waiting on an Activity Level, is only a few milliseconds.

4 (EOR) End of Resource Utilization Time:

The application program completes using system resources. This normally coincides with the End of Transaction (EOT).

At this point, the program has performed an operation that causes workstation I/O to send data to the display. The following user program operations cause the data to be sent to the display:

- Read or invite input operation following one or more output operations with DFRWRT(*YES) in the display file description
- Output operation with DFRWRT(*NO) in display file description
- Output operation with the DDS INVITE keyword function
- Combined output/input operation (for example, EXFMT in RPG/400 and SNDRCVF command in Control Language programs)
- End of program

5 (EOT) End of Transaction:

This is the end of the System Measured Response Time. The next transaction may begin. Resource usage by the transaction is measured at this point. This may coincide with the End of Resource Utilization Time (EOR). Any Active-Wait transition is included here.

6 System response is displayed to user.

D.1.1 Start and Stop Points Of Steps Within a Transaction

1 -> 6 Display I/O Transaction Path:

The complete path taken by the transaction from the time the user presses the Enter key or a Function key to the time when a response is received by the user. This equates to the workstation user′s perception of response time.

2 -> 3 Application Input Queuing Time:

This is the time the input data waits before the system resources are made available to it. Examples are waiting on:

- An activity level
- The program to issue an input operation
- The program to accept input

3 -> 4 Transaction Resources Usage:

The period when system resources are used for processing. It includes periods of waiting, such as object seize/lock conflicts and resource queuing.

2 -> 5 System Response Time

1 -> 2 Components of response time not recorded by the system

5 -> 6 Components of response time not recorded by the system

Appendix E. OS/400 Expert Cache and Set Object Access Overview

This section provides more in-depth information regarding the user-specified ″pinning″ of AS/400 objects or portions of objects into main memory storage pools. The following topics are discussed:

- Set Object Access (SETOBJAC command) support
- Expert Cache support

E.1 Set Object Access

Set Object Access is a command introduced to enable the user to load ″data″ into a specific pool of memory. The pool can either be a shared pool or a private pool. The ″data″ can be a database file, database index (logical file), or a program. The object can be directed into the specified pool and remain there to eliminate any disk I/Os from accessing the data at run time.

While programs can be ″pinned″ in main storage with the SETOBJACC command, the support is more commonly used with a physical database file or the file index.

SETOBJACC typically can cause the biggest performance improvement for applications with heavy random access to database files. In these jobs, random access to physical and logical files can cause much longer run time than does sequential access to the same data **because almost every program access to the data requires a physical disk I/O operation (key or record access)**. Sequential processing files may also benefit but in most cases, standard system support for sequential processing or using *expert cache* is sufficient and easier to use than SETOBJACC.

Before using SETOBJACC, you **must** use the Clear Pool (CLRPOOL) command first. This command gets rid of everything in the pool and is not interruptible. CLRPOOL writes all changed pages in a pool to disk and indicates that all pages in a pool are empty. Upon job completion, the pool should be cleared using the CLRPOOL command to ensure that objects are purged out to disk. You can also do this during the job′s run time (for files that are no longer being used) by using the SETOBJACC *PURGE option to clear them from main storage.

To ensure that SETOBJACC gives you the performance improvements you expect, you must have an understanding of which programs and files (both logical and physical) are used by your job during run time. You must also understand what other objects are loaded into that same storage pool based on the compare values on the subsystem monitor′s routing entries.

SETOBJACC is ″pool specific″. Once the command is issued for an object, the object stays in the pool until you clear the pool, issue a SETOBJACC command specifying POOL(*PURGE) for the object, or issue another SETOBJACC for the **same object/same pool**.

If you specify loading the same object/same pool again, the ″previous copy″ of the object is purged and the object is reloaded. Any program currently using the ″previous copy″ is impacted and, of course, the job issuing the SETOBJACC has to wait for the purge and the reload.

Note: Do not put CLRPOOL and SETOBACC commands for the same storage pool in an initial program run from many workstations. Each time the commands are issued, it causes the pool to be cleared and the object ″re-loaded″.

Other important factors to keep in mind when deciding to use SETOBACC support include:

- Depending on the size of the storage pool and the number of changed pages within the pool, clearing the storage pool through CLRPOOL can take considerable time.

  The CLRPOOL command removes everything from a storage pool. The command writes all changed pages to disk and indicates that all pages in the pool are empty. The first time a user invokes the CLRPOOL command on a storage pool, it may run for a long time, depending on the number of changed pages that are written. The long run time is usually after starting the subsystem to define the pool.

  The CLRPOOL command is not interruptible. Once it starts, it does not check to see if a user wants to cancel it. The command runs until the pool has been cleared.

  After initially using CLRPOOL, it is faster to explicitly purge an object using the SETOBJACC command rather than the CLRPOOL command. The SETOBJACC command passes a list of object addresses to storage management. Storage management can efficiently locate pages of the objects and asynchronously block writes to disk. The CLRPOOL command causes storage management to go through the pool one page at a time and synchronously write each changed page.

- Make sure the pool is large enough to contain the entire object.

  Objects are loaded from first byte to last byte. If the storage pool has insufficient storage, the last ″n″ bytes of the object are in main storage at the completion of the SETOBJACC command.

  See Section E.1.2, “SETOBJACC - Determining Object Size” on page 415 for information on object size determination and sample SETOBJACC messages.

- Deleted record space is loaded into main storage.

  Consider reorganizing the file (RGZPFM command) or re-using deleted records (CRTPF command REUSEDLT parameter) before file usage with SETOBJACC.

- Data added to a ″pinned″ file uses space in the *job storage pool*, not the pool used to pin the data.

- There is no command to display which objects are located in a storage pool.

E.1.1 SETOBJACC - Determining Which File Should Be Used

The system provides no direct information on deciding which files gain the most by pre-loading with SETOBJACC, such as a count of the physical I/O operations per file. Logical disk I/O operations per file do not represent the physical I/O operations to the disk but the logical I/O count may give you an indication of which files have a high disk I/O activity.

One way to get the count of logical I/Os per file is to use the Performance Monitor while running important applications.

Use the Performance Tools/400 (option 70) Display Access Group (DSPACCGRP) command over the jobs you are interested in. Use the Analyze Access Group (ANZACCGRP) command′s file report to see which files have the highest I/O counts.

You can also try determining this ″real time″ by using the OS/400 Display Job (DSPJOB) command option 14, Display Open Files. The output shows the logical I/Os issued by the job during the snapshot of time the DSPJOB command reports.

Always consider that both ANZACCGRP and the DSPJOB open file output are snapshots of system performance only and must be considered against the entire application run time.

E.1.2 SETOBJACC - Determining Object Size

As previously stated, the full performance benefit of SETOBJACC requires sufficient storage to contain the entire object.

For database files, you can approximate this by using the Display Physical File Description (DSPFD) command′s TYPE = *MBRLIST parameter to get the member size and add 30KB to 40KB to get approximate actual size.

For a program, use Display Object Description (DSPOBJD) with the *FULL option.

In all cases, you are approximating the size of the object. You need to review the SETOBJACC command messages such as CPC1140 for a file and CPC1141 for a program to determine the actual size of the object. These messages indicate the size of the object and the amount of pool storage available *before* the object load is started. If the amount of pool storage available is *less than* the object size shown in the message, using SETOBJACC may not achieve maximum performance.

Consider the following example messages for program COMPTIME and file CSTFIL.

```
> SETOBJACC OBJ(CMN38/COMPTIME) OBJTYPE(*PGM) POOL(CHAINBCH 1)
  7KB of COMPTIME brought to pool with 4000KB unused.

> SETOBJACC OBJ(CMN38/CSTFIL) OBJTYPE(*FILE) POOL(CHAINBCH 1)
  2019KB of CSTFIL brought to pool with 3993K unused.
```

SETOBJACC for COMPTIME was issued immediately after the CLRPOOL command completed for pool 1 in subsystem CHAINBCH. Pool 1 is initially set to 4000KB (4MB). After COMPTIME is successfully loaded, 3993K remains available for CSTFIL, which is 2019KB in size.

If you try to load an object that is larger than 1974KB in size, accessing that object may not achieve maximum performance.

E.2 Expert Cache

Expert cache is a selectable option under OS/400 that enables the system single-level storage support to use main memory as a cache. Expert cache is specified at the **shared storage pool** level and overrides standard storage management and database management algorithms for keeping objects or portions of objects available in main storage. The objective of expert cache is to significantly reduce the number of physical disk I/O operations without the more detailed knowledge of file and program usage required by the SETOBJACC command support.

Expert cache can be turned on for a shared storage pool by specifying *CALC for the PAGING parameter on either the Change Shared Storage Pools (CHGSHRPOOL) command or the Work with Shared Pools command menu. By specifying *FIXED, expert cache is turned off for the specified pool. It is beneficial that the user understand what jobs are running in the storage pool when specifying *CALC. If *CALC does not result in improved performance, simply change the pool back to *FIXED or modify your subsystem monitor routing entries to more closely control which jobs are affected by expert cache.

OS/400 APIs (QUSCHGPA, QWCCHGTN, and QWCRSSTS) enable changes to storage management algorithms for object page management. These APIs are not recommended for the average system user and should be used by those who are knowledgeable on how expert cache works.

Expert cache can be contrasted to the SETOBJACC support where the user must identify which objects should be loaded into main storage and must consider that there is sufficient main memory to load the entire object into main storage. Also, expert cache periodically analyzes the object reference patterns and can determine when new objects or portions of objects should be cached because of changes in the application environment over time.

Performance improvements can be achieved in various workload environments but typically are realized where data is physically contiguous within memory. This is termed ″locality of reference″ under expert cache support and objects or portions of objects with a high locality of reference are candidates for caching.

Batch applications usually see improvements in run time.

In some cases, using the Override Database File (OVRDBF) command SEQONLY(*YES number-of-records) parameter can approximate the performance improvement under expert cache, but this also requires the user to predetermine which files to select and is more cumbersome to use if an application environment changes over time.

E.2.1 Expert Cache - Algorithm Details

Expert cache directs the system to determine which objects or portions of objects should remain in a shared main storage pool based on the reference patterns of data within the object. Expert cache overrides the standard AS/400 storage management algorithm described in Section 3.9.2, “Memory (Storage) Pools” on page 36.

Expert cache uses a storage management tuner that runs independently of the system dynamic tuner to examine overall paging characteristics and history of the pool. The tuner can be seen as active from time to time by repetitively refreshing the Work with System Activity (WRKSYSACT) command display and noting task SMXCSPRVSR.

Expert cache dynamically changes the way pages are managed based on pool activity and object usage. The storage management tuner task tells OS/400 database:

- How large disk reads (brings) should be for a particular file class.

- Whether or not we should do Exchange Brings.
- Whether we should write changed pages to disk, purge them, or do nothing.

Files with a higher locality of reference (″hot spots″) tend to stay resident if enough memory is available.

The theory behind expert cache is to ″manage″ the database pages working set size in a pool to be as large as possible without causing excessive paging of non-database pages. Performance should improve as the number of physical I/Os performed per transaction decreases.

Once expert cache classifies the object access behavior, it acts on the object by varying the transfer size and by ensuring that highly used database data stays in memory longer. When a job can effectively use more than one page of data (4KB), expert cache uses data blocking to retrieve the data from disk.

Depending on the amount of data the job uses, expert cache varies the blocking factor from 4KB up to 128KB for database objects.

The storage management tuner examines overall paging characteristics and history of the pool to adjust the values in its tuning table for that pool. Periodically, the tuner wakes up and tunes each pool that has a paging option of *CALC. As well as tuning the pool, it may also reclassify a file into a new class, if necessary, based on size of the file and if the access history for the file has changed from the previous analysis. The steps involved for each pool include:

- Calculate the job′s share of the pool.
- Calculate number of transactions within the pool by computing internal time slice ends plus active to wait machine interface transitions.
- Gather pool I/O counts.
- Calculate I/Os per transaction.
- Cycle pool through tuning states.
- Age pool:

  If changed pages are not getting recycled frequently enough, write the oldest changed pages to disk.
- Call the ″file classer″.

Each Data Space and Data Space Index has a different classification for each pool. There are three classes defined to the storage management tuner and one class defined by database:

- Class 1 (file access appears to be random):

  A disk access is required for nearly each record that is accessed.
- Class 2 (some locality of reference is detected):

  Several records are being accessed per disk access.
- Class 3 (high locality of reference detected):

  The file is being processed in a sequential manner, references are highly clustered, or large portions of the file are resident in memory.
- Class 4 (special file treatment):

  Database management can determine the file is small enough and available storage is sufficient to load the entire object into main storage.

All non-database object paging is treated on an ″as-required″ basis (dependent on object type), where storage management determines how much of that object needs to be brought into the pool (anywhere from 4KB bytes to 32KB). Non-DB objects include data queues, data areas, spooled files, programs, and so on.

At IPL time, the tuner classifies all database files into Class 2 as the default, regardless of whether Expert Cache is turned on or off.

It is helpful if you organize your data according to the most-used keys and keep your files ″clean″ by removing deleted records as the deleted space makes expert cache less efficient.

If you have multiple Open Data Paths (ODPs) per job, each ODP may have a different access method, but the file still has one classification per evaluation period.

Storage management supports a ″sweeper task″ that ages pages in the storage pool. When started, the task monitors the page demand rate in a pool. This rate is the number of non-purged pages allocated from the pool (the rate the pool is being ″turned over″). If this rate, over some period of time, is less than twice the size of the pool, the ″sweeper task″ initiates sufficient aging activity in the pool to reach that rate.

Note that with the capability to have large memory pools, the risk of data loss (data staying in memory longer) for non-journaled files has increased over smaller system hardware configurations. You must always consider disk mirroring, RAID-5, uninterruptible power service hardware, battery backup features, physical file and access path journaling, and even database force-write-ratio of 1 to minimize loss of data due to unanticipated hardware or power failures.

A shared pool paging option other than *FIXED may increase the exposure to data loss if these recovery procedures are not in place, though the sweeper task is designed to minimize this risk.

Appendix F. Overview of UNIX Work and Resource Management Constructs

V3R6 provides additional SPEC 1170 enhancements over V3R1 giving greater system openness and using Integrated Language Environment (ILE) capabilities to support a variety of UNIX-based applications.

With the AS/400 Common Programming API (CPA) Toolkit (an orderable feature of OS/400), new AS/400 applications can be written using UNIX-based constructs with ILE C/400. These applications can use system interfaces and C run-time functions compatible with OS/2, DOS, NT, POSIX, XP, and UNIX:

- POSIX compatible threads
- SVID compatible semaphores
- SVID compatible shared memory
- Thread-safe ANSI C I/O run time
- Thread-safe BSD compatible sockets
- Thread-safe BSD compatible network run time
- Miscellaneous SVID or BSD compatible C run-time functions

  SVID is the System V (five) UNIX operating system originally developed by AT&T. BSD is the (University of California) Berkeley Software Distribution UNIX operating system.

Extensive discussion of performance management for UNIX-based applications is beyond the scope of this redbook. However, this section provides an overview of the UNIX-based constructs for work management and resource management for the performance management expert who is involved with porting UNIX-based applications to OS/400.

Detailed experience with performance management of a UNIX-based application is addressed in future performance documentation.

UNIX work management and resource management constructs discussed in this section include:

- Shell
- Kernel
- Processes
- Forks
- Pipes
- Message Queues
- Semaphores
- Shared Memory
- Signals
- Threads
- Mutexes

F.1.1 Shell

A *shell* is a command processor that is the actual interface between the user and the UNIX ″kernel″. The range of functions (commands) available through a shell is dependent upon the shell developer, such as the (Stephen) *Bourne shell*, the *C-shell* (developed by William Joy of the University of California at Berkeley), the *Korn shell* (distributed by AT&T), and so on.

F.1.2 Kernel

The UNIX *kernel* typically includes the UNIX operating system support for work scheduling, memory management, device drivers and various ″system built-in″ functions. These functions are accessed by *system calls.* Some UNIX operating systems have different ″extensions″ to a standard set of kernel functions shared across most UNIX operating systems.

F.1.3 Processes

The activity that occurs when a UNIX command is running is called a *process.* In the UNIX environment, multiple processes can be running simultaneously. Multiple processes can be started from the same workstation. The UNIX ″kernel″ assigns a unique address-process ID to each process. A process can be in the ″foreground″ (in a workstation dialogue) while others are running in the ″background″.

This process support is similar to having an interactive job do multiple AS/400 ″submit job″ commands. The foreground process can communicate with the workstation operator, while the background processes cannot communicate directly to the workstation operator. The processes started and the original process that started the other processes are part of a *process group*.

The Performance Monitor (and Performance Tools/400) indicates a UNIX process with ″BCI″ (Batch-Communication-Interactive) as a job type.

Many of the following UNIX constructs are oriented to inter-process communication.

F.1.4 Forks

A *fork* is a system call that is used to clone a new process from an existing process. The process that issued the fork system call is called the parent process whereas the newly initiated process is called the child process.

Any files already opened in the parent process before the cloning are also available to the child process after the fork.

After the fork system call has completed, the parent can terminate access to the files so that the parent and child processes are not sharing the file.

Note that the relationship between parent and child processes are not supported under V3R6 OS/400.

F.1.5 Pipes

A *pipe* is one of the ″objects″ that can be used to permit two processes to exchange data. The pipe represents a connection between two processes. The output of one process becomes the input to the other process through the pipe. A typical pipe usage flow can be as follows:

- Process A1 creates a pipe and uses the fork to create a copy of itself (A2).
- The parent process (A1) places data into the pipe and closes the read end of the pipe.
- The child process (A2) reads the data from the pipe and processes the data. The child process cannot put data onto the same pipe.

  The pipe provides a one-way flow of data between the two processes.

A pipe is uni-directional. To have two-way communication between processes through pipes, two pipes are required, one input and one output for each process.

F.1.6 Messages

A *message queue* is one of the ″objects″ that can be used to communicate between processes. A message queue can be compared to a mailbox and the message compared to a letter in the mailbox.

It is possible to have a process write to a message queue and exit, which means the message queue to exchange data can be processed asynchronously at a later time.

For a pipe, both the read and write processes have to exist at the same time. For a message queue, the sending process and receiving process can exist at separate times.

UNIX message queue support is similar to standard AS/400 message queue support, but with its own unique interfaces.

F.1.7 Shared Memory

Shared Memory can be a pipe, a message queue, or a file that is declared as a shared memory segment. Shared memory segments must be processed one at a time and a semaphore can be used to indicate whether the shared memory segment is in use or can be used by another process.

The shared memory can be used similar to a standard AS/400 data area that contains the next invoice number where each job locks the data area, updates the current invoice number, and unlocks the data area for other jobs to use the new invoice number.

F.1.8 Semaphores

A *semaphore* is a kind of ″signal″ to another process that something has been completed. It is not designed for exchanging large amounts of data and is typically used to synchronize usage of *shared memory*.

For example, a semaphore is defined for file FILS. Process A1 and Process B1 want to access the shared memory segment (FILS). Process A1 is using FILS. When finished, A1 sets the semaphore to ″1″, which means FILS is available.

Process B1 checks the semaphore for ″1″. When the semaphore value is ″1″, Process B1 can access the file.

F.1.9 Signals

A *signal* is a notification to a process that ″something″ has occurred. Signals can be sent:

- One process to another
- By the ″kernel″ to a process

Signals can indicate some hardware or software condition. For example, a signal is a way to tell a process that it should terminate itself.

Signals can be processed immediately by the process receiving the signal or the process can take note of the signal and handle the signal at a later time.

F.1.10 Threads

A *thread* is a sequential flow of control within a process. A single process can have multiple threads. When a process has multiple threads, it is called a multi-threaded process.

A multi-threaded process can have multiple flows of execution with all threads within the process sharing the same address space.

Most applications are designed to have a single flow of control within a single thread process. More sophisticated applications can be developed that have:

- Multiple single thread processes:

  The processes occupy different address spaces but can communicate through pipe, message queues, and shared memory.

- A multiple thread process:

  Multiple thread processes have multiple flow of control *within the same process*. All threads have access to the same address space.

In both the multiple single thread processes and the multi-thread single process designs, the programmer is responsible for determining which functions can be placed into one thread while another thread continues to do work independent of the other thread. One application example is for an order entry thread to continue processing an order from workstation I/O while another thread is determining tax rules that have to be applied when the order has been completed.

This kind of design is implemented with ″non-UNIX″ constructs such as message queues, data queues, data areas, and the OS/400 Allocate Object (ALCOBJ) command and Deallocate (DLCOBJ) command. However, UNIX constructs for sharing data and synchronizing use of the data provide much more sophisticated interfaces for this application design.

In most multi-process or multi-thread designs, the programmer is responsible for managing the threads and synchronizing their use of global (across all process or threads) resources.

There are several UNIX-based terms and capabilities to assist this management and synchronization that include semaphores, signals, mutexes, threading attributes (such as ″thread-safe″), and so on.

When an AS/400 CPA program is called, the system creates and initializes control structures that enable a process to become multi-threaded. The OS/400 job in which the CPA program is called becomes the primary or initial thread if the process becomes multi-threaded. Secondary threads for the process are created through the ″pthread_create()″ function. When the pthread_create() function completes, the thread ID of the secondary thread is returned to the primary thread job for later use, if necessary.

On the AS/400 system, the secondary thread becomes another job with a ″fast path″ through job initiation. These new jobs have the same library list, user profile, and run attributes as the job with the initial primary thread. CPA support creates the additional UNIX control structures required beyond those generated for ″normal″ AS/400 jobs.

On the Work with Active Jobs (WRKACTJOB) command display, a secondary thread is indicated by a job type of ″BCI″ (batch immediate).

F.1.11 Mutexes

A *mutex* (**mut**ual **ex**clusion) is a mechanism for synchronization between threads. Mutexes provide a fast interface to serialize access to shared global storage.

If not used properly, they can cause deadlock situations. CPA does not detect deadlock conditions.

When a thread is waiting on a mutex, the WRKACTJOB command job status indicates ″MTXW″. DSPJOB has an option for displaying the mutexes held by a job/thread, which mutex the job is waiting for, and which thread, if any, holds the mutex being waited for.

Mutexes are not automatically destroyed or released when a thread ends. Mutexes are destroyed when the storage from which they were allocated is returned to the system for re-use, such as when the original CPA program ends.

F.1.12 UNIX-Based Work Management Summary

V3R6 and the AS/400 CPA Toolkit provide many UNIX-based application functions and interfaces that include work management and resource management. These capabilities can be used in sophisticated application designs and managing them requires both AS/400 system and UNIX skills in work management and resource management.

For more details, refer to *Common Programming APIs Toolkit/400 Reference Version 3*, SC41-4802.

Appendix G. OS/400 Client Access/400 Subsystem and Job Information

Client Access/400 jobs run in either subsystem QCMN or QSERVER. The old PC Support subsystem QXFPCS is no longer available.

Table 50 shows the V3R6 QCMN subsystem routing entry compare values for V3R6 Client Access/400 host server programs. Most, but not all of the routing entries are shipped with the V3R1 release. The function description notes when a subsystem routing entry is **not shipped with the V3R1 software release.**

Table 50 (Page 1 of 2). V3R6 QCMN Client Access/400 Routing Entries

Seq Number, Library	Compare Value	Compare Start Position	Job Name	Class	Function
10 QIWS	′QZSCSRVR′	37	QZSCSRVR	QCASERVER	Central server for license management, client management, and so on.
20 QIWS	′QZRCSRVR′	37	QZRCSRVR	QCASERVR	Remote command, distributed program call requests
30 QIWS	′QZHQTRG′	37	APPC device	QCASERVR	Data queues (OS/2 client)
50 QIWS	′QVPPRINT′	37	APPC device	QWCPCSUP	Virtual print (Original, Windows 3.1 clents)
60 QSYS	′QNPSERVR′	37	QNPSERVR	QCASERVR	Network print server (OS/2 client)
70 QIWS	QNMAPINGO′	37	APPC device	QWCPCSUP	
80 QSYS	QNMAREXECD′	37	APPC device	QWCPCSUP	
100 QIWS	QTFDWNLD′	37	APPC device	QWCPCSUP	All file transfer requests (All clients)
150 QIWS	′QMFRCVR′	37	APPC device	QWCPCSUP	Message function receiver
200 QIWS	′QMFSNDR′	37	APPC device	QWCPCSUP	Message function sender
210 QIWS	′QHQTRGT′	37	APPC device	QWCPCSUP	Data queue requests (Original and Windows 3.1 clients)
220 QIWS	′QRQSRV′	37	APPC device	QWCPCSUP	Remote SQL servers - QRQSRVX, QRQSRV0, QRQSRV1. See note 4. Note that ′QRQSRV′ includes all three.
240 QIWS	′QLZPSERV′	37	APPC device	QWCPCSUP	License management requests (Original clients and Windows 3.1 clients)
250 QSYS	′QCNPCSUP′	37	APPC device	QWCPCSUP	Remote command, shared folders 0,1 (Pre-V3R1 clients)
260 *RTG	′QOCEVOKE′	37	APPC device	QINTER	Used by OV/400 for cross-system calendar
290 *RTG	′QPCSUPP′	01	APPC device	QBATCH	All programs not satisfying a previous compare value but using APPC mode QPCSUPP initiate a program in QCMN. This is used by PC Support/400 clients to initiate WSF 5250 sessions.
295 *RTG	′QCASERVR′	01	APPC device	QCASERVR	All programs not satisfying a previous compare value but using APPC mode QCASERVR initiate a program in QCMN. This is a Client Access/400 APPC mode.
299 *RTGDTA	′#INTER′	01	APPC device	QINTER	All programs not satisfying a previous compare value but using APPC mode #INTER initiate a program in QCMN.

Table 50 (Page 2 of 2). V3R6 QCMN Client Access/400 Routing Entries

Seq Number, Library	Compare Value	Compare Start Position	Job Name	Class	Function
300 *RTG	′PGMEVOKE′	29	APPC Device	QBATCH	General program start request processing. See Note 1.

Note:

Notes

1. Routing entry sequence number 310 is a ″catch all″ for all incoming program start requests not satisfying a previous routing entry comparison. Jobs processed by this entry include user-written applications and all ″pass-through″ functions (5250 Display Pass-through, TELNET, Work Station Function (WSF), RUMBA, PCS5250).

   The actual interactive job of the pass-through defaults to run in subsystem QINTER.

2. Note that Client Access/400 host program QSYS/AACSOTP (APPC sign on transaction program) has no specific routing entry compare value.

3. All class descriptions listed assign jobs to pool 1 (BASE pool).

   Since there are many jobs from a wide variety of applications such as Client Access/400, ADSM/400, Pass-through, Distributed Data, customer applications, and so on, you should consider adding storage pool (or pools) and routing jobs within the same application into a pool other than *BASE.

4. Different Remote SQL programs are called, depending on the client PC Support/400 or Client Access/400 support:

   - QIWS/QRQSRVX: V2R2 and later PC Support/400 clients and Client Access/400 original clients and Client Access/400 for Windows 3.1.
   - QIWS/QRQSRV0: Pre-V2R2 PC Support/400 clients using commitment control *NONE.
   - QIWS/QRQSRV0: Pre-V2R2 PC Support/400 clients using commitment control *ALL.

Note that all file server jobs are first routed to QCMN subsystem and internally moved to QSERVER subsytem.

Subsystem QCMN is shipped with a prestart job entry for program QNPSERVER QOQSESRV, QZRCSRVR, and QZSCSRVR. You should review the prestart job entries for Client Access/400 functions to determine if their default parameter settings permit maximum efficiency for the functions being used. For example, all Client Access/400 servers (QNPSERVER, QZRCSRVR, QZSCSRVR) only have one job prestarted initially.

Note that the Windows 3.1 clients using Remote SQL perform their work in the job name associated with the attached client APPC control unit description. You can use this information to identify AS/400 CPU and disk resource utilization for each client. Also, there may be multiple APPC jobs active concurrently with each client, depending on the Client Access/400 functions used.

Use the subsystem information to control Client Access/400 function/job run priority and storage pool assignments and to group jobs for performance measurement and BEST/1 capacity planning.

Table 51 lists the routing entry compare values for QSERVER subsystem that provides all Client Access/400 file server and database server jobs.

Table 51 (Page 1 of 2). V3R6 QSERVER Client Access/400 Routing Entries

Seq Number, Program, Library	Compare Value	Compare Start Position	Job Name	Class	Function
100 QPWFSERV - QSYS	′QSRVR′	01	QSERVER	QPWFSERVER	Main file server program, one job for each client (all clients)
200 QPWFSERVSD - QSYS	′QPWFSERVSD′	01	QSERVER	QPWFSERVER	
300 QPWFSERVER - QSYS	′QSTART′	01	QSERVER	QPWFSERVER	QSERVER auto start job Compare value is APPC mode.
400 QZDAINIT - QIWS	′QZDAINIT′	37	QZDAINIT	QPWFSERVER	Transaction program for database servers (Windows 3.1 ODBC, for OS/2 client ODBC, Remote SQL, and new file transfer APIs). See Note 1.

Table 51 (Page 2 of 2). V3R6 QSERVER Client Access/400 Routing Entries

Seq Number, Program, Library	Compare Value	Compare Start Position	Job Name	Class	Function
600 QZDASRVSD - QIWS	′QZDASRVSD′	37	QZDAINIT	QPWFSERVER	Transaction program for database servers (Windows 3.1 ODBC, for OS/2 client ODBC, Remote SQL, and new file transfer APIs). See Note 1.

Note:

Notes.

1. Program QIWS/QZDANDB processes native database requests. Program QIWS/QZDAROI processed information requests for database files and SQL catalog functions. Program QIWS/QZDASQL processes SQL requests. You may consider specific routing entry compare values for these programs to change their default priority or storage pool.
2. Routing entry 500, which also compares for QZDAINIT starting in position 37, was shipped with V3R1 subsystem QSERVER. It is considered redundant and will be removed at a later time.

Subsystem QSERVER is shipped with a prestart job entry for program QZDAINIT. You should review the prestart job entries for Client Access/400 functions to determine if their default parameter settings permit maximum efficiency for the functions being used.

All prestarted QZDAINIT jobs use user profile QUSER. When a specific client attaches to a QZDAINIT job, the server code places a message in the job log that identifies the currently running client. This message exists in the job log until the client detaches. Each prestarted QZDAINIT job may attach to multiple clients over time. You can use this information to identify AS/400 CPU and disk resource utilization for each client, but this dynamic connecting and disconnecting from multiple clients is not preserved in any Performance Monitor data.

Note that the Windows 3.1 clients using ODBC and OS/2 optimized clients using either ODBC or Remote SQL attach to a QZDAINIT prestart job.

The Performance Monitor data for the QZDAINIT jobs collects CPU and disk utilization data. The number of communications I/Os are not included since they are actually performed by LIC tasks.

Use the subsystem information to control Client Access/400 function/job run priority and storage pool assignments and to group jobs for performance measurement and BEST/1 capacity planning.

G.1.1.1 Client Access/400 User Exits: Client Access/400 provides user exits that enable user customized processing at key ″registered exit points.″ The user exit programs may do what ever function the customer wants within the limits of the exit points supported by Client Access/400.

V3R6 Client Access/400 provides exit points through the OS/400 Registration Facility.

From a performance viewpoint, the V3R6 exit points enable fewer unnecessary calls to a user exit program. See *OS/400 Server Concepts and Administration Version 3*, SC41-3740, for more information.

Appendix H. Special Notices

This publication is intended to help customers, IBM system engineers, and IBM business partners implement the structures for performance management on the AS/400 system. It contains a description of the methodology for performance management and an ordered procedure for using the basic AS/400 performance tools.

The information in this publication is not intended as the specification of any programming interfaces that are provided by the AS/400 Performance Tools, 5716-PT1. See the PUBLICATIONS section of the IBM Programming Announcement for the AS/400 Performance Tools, 5716-PT1, for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR
Advanced Function Printing
Advanced Peer-to-Peer Networking
Advanced 36
AFP
AIX
AIX/6000
AnyNet
Application System/Entry
Application System/400
APPN
AS/400 PerformanceEdge
AS/400
BookManager
BookMaster
C Set ++
C/400
CallPath/400
CICS/400
COBOL/400
Common User Access
CUA
Current
DATABASE 2 OS/400
DataPropagator
DB2/400
DRDA
Facsimile Support/400
GDDM
IBM
IBMLink
IIN
ImagePlus
InfoWindow
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
OfficeVision/400
Operating System/400
OS/400
PowerPC
PowerPC Architecture
PowerPC AS
PROFS
PSF
RPG/400
SNAP/SHOT
SQL/400
System/36
System/38
SystemView
ThinkPad
Ultimedia
VisualAge
VisualGen
VRPG CLIENT
YEAR 2000

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

Appendix I. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

I.1 International Technical Support Organization Publications

These publications are relevant as further information sources:

- *AS/400 Performance Explorer - Tips and Techniques*, SG24-4781
- *AS/400 Client/Server Performance Using the Windows Clients*, SG24-4526-01
- *AS/400 Client/Server Performance Using Application Development Tools*, SG24-4731
- *Database Parallelism on the AS/400*, SG24-4826
- *AS/400 Communication Performance Investigation*, SG24-4669
- *AS/400 Performance Management V3R1*, GG24-3723-02
- *Moving to ILE for RPG IV*, GG24-4358

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 435.

I.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	**Subscription Number**	**Collection Kit Number**
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

I.3 IBM AS/400 Related Publications (White Books)

These publications are relevant as further information sources:

- *OS/400 Work Management Version 3*, SC41-4306
- *Performance Tools/400 Version 3*, SC41-4340
- *AS/400 Performance Tools/400 - Getting Started V3R6*, SC41-4343
- *DB2 for AS/400 Database Programming Version 3*, SC41-4701
- *DB2 for AS/400 SQL Programming Version 3*, SC41-4611
- *DB2 for AS/400 SQL Reference Version 3*, SC41-4612

I.4 Other Publications

The following articles are contained within the identified issue of *NEWS 3X/400*, a non-IBM magazine issued monthly that contains articles contributed by both IBM and non-IBM authors. While all of these articles were written by Rick Turner, formerly with the US AS/400 Competency Center in Rochester, Minnesota, various issues of *NEWS 3X/400* may contain other articles on the AS/400 system.

- *″AS/400 Performance Workshop: Part 1 Expert Cache,″ NEWS 3X/400*, May 1994.
- *″AS/400 Performance Workshop: Part 2 Cache In on the AS/400,″ NEWS 3X/400*, July 1994.

  Much of the Set Object Access support discussed in Appendix E, "OS/400 Expert Cache and Set Object Access Overview" on page 413 is based on this article.
- *″Share Data Faster Between AS/400′s OptiConnect/400,″ NEWS 3X/400*, September 1994
- *″Identify AS/400 Resource Hogs (TPST)″*, *NEWS 3X/400*, November 1994
- *″Tune Your Programs with SAM″*, *NEWS 3X/400*, January 1995
- *″Under the Covers of AS/400 Memory Management″*, *NEWS 3X/400*, March 1995
- *″Under the Covers of AS/400 CPU Management″*, *NEWS 3X/400*, May 1995

Another useful publication is:

- *Mastering AS/400 Performance*, by Rick Turner and Alan R. Arnold, Duke Press 1996.

The *AS/400 Magazine* produced by the US Competency Center in Rochester, MN, U.S.A., often has articles discussing performance and should be considered another good source of timely performance information.

The following publication is an internal use only document. Only IBM systems engineers and business partners are authorized to obtain this document. The SE and business partner are responsible for determining what information from this document may be shared with a specific customer.

- *AS/400 Performance Capability Reference for PowerPC Technologies V3R6*, ZC41-0607

  This document contains internal laboratory performance test results that are helpful in setting reasonable expectations and understanding the relationship between system resources and certain system parameter values.

  To ensure you have access to the latest copy of this manual, we recommend using the following VM command to obtain a copy:

  ```
  TOOLCAT MKTTOOLS GET AS4PPCPF PACKAGE
  ```

  The *System Support Diagnostic Aids (Volume 1)*, LY44-0597, (available to IBM licensed customers only), lists Licensed Internal Code (LIC) tasks.

I.5 Other Documentation

HONE Electronic Question and Answer (Q&A) support includes several item that discuss AS/400 performance. Several key items reference **internal IBM items** that can be used to set realistic performance expectations:

- Item RTA000089352

  This item summarizes performance-related PTFs and is continually updated. This item should be reviewed monthly for PTFs applicable to various customer environments.

- Item RTA000011842

  This item summarizes performance considerations and is generally updated at key times, such as an AS/400 announcement or general availability date. It is also an index to specific items that contain more detail on a particular subject, such as Database, Client Access/400, LAN Server/400, and so on.

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL `http://www.redbooks.ibm.com`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type `GOPHER.WTSCPOK.ITSO.IBM.COM`
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  `http://w3.itso.ibm.com/redbooks`

- **IBM Direct Publications Catalog on the World Wide Web**

  `http://www.elink.ibmlink.ibm.com/pbl/pbl`

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to `announce@webster.ibmlink.ibm.com` with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
(+45) 48 14 2207 (long distance charge)	Outside North America

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number ____________________

- Credit card number ____________________

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Abbreviations

ABBREVIATION	MEANING
APMT	Automated Performance Management Tool
APPC	Advanced Program to Program Communication
CL	Control Language
CPYF	Copy File Command
CPI-C	SAA Common Programming Interface - Communications
CPU	Central Processing Unit
DDM	Distributed Database Management
EAO	Effective Address Overflow
FSIOP	File Server I/O Processor
FWR	Force Write Ratio
HLL	High Level Language
ICF	Inter System Communication Facility
IFPS	Integrated PC File Server
ILE	Integrated Language Environment
IMPI	Internal Microprogramming Interface
IOP	Input Output Processor
ITSO	International Technical Support Organization
LIC, VLIC	Licensed Internal Code (software and hardware)
MRT	Multiple Requesting Terminal
OCR	Over Commitment Ratio
OEM	Other Equipment Manufacture
OPM	Original Program Model
PAG	Process Access Group
PEX	Performance Explorer
PRPQ	Programming Request for Price Quotation
SAM	Sample Address Monitor
SLIC	System Licensed Internal Code
TPST	Timing and Paging Statistics
WSF	Workstation Function

Index

ITSO Redbook Evaluation

AS/400 Performance Management V3R6/V3R7
SG24-4735-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

____

____

____

____

What other redbooks would you like to see published?

____

____

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

____

____

____

____

____

Printed in U.S.A.

SG24-4735-00