International Technical Support Organization
AS/400 Performance Management Version 3 Release 1
April 1995

Document Number GG24-3723-02

April 1995

International Technical Support Organization
Rochester Center

Take Note!

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

Third Edition (April 1995)

This edition applies to Version 3, Release 1, Modification 0 of the AS/400 Operating System and Performance Tools licensed program Program Number 5763-PT1.

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

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

IBM Corporation, International Technical Support Organization
Dept. 977 Building 663-3
3605 Highway 52N
Rochester, Minnesota 55901-7829

Abstract

This document describes a methodology for performance management on IBM AS/400. 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.

The Version 3 Release 1 Performance Tools (5763-PT1) reports and the Advisor function are the primary tools used for performance management. Other performance evaluating tools are also discussed.

This document is intended mainly for IBM system engineers and IBM business partners who want to implement a performance management structure on an AS/400 customer system, but may also be useful to IBM customers.

An intermediate knowledge of the Performance Tools licensed program (5763-PT1) is assumed.

(500 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help customers, IBM system engineers and IBM business partners to implement the structures for performance management on AS/400. 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, 5763-PT1. See the PUBLICATIONS section of the IBM Programming Announcement for the AS/400 Performance Tools, 5763-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.

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	AFP
AIX	AIX/6000
Ami Pro	AnyNet
Application System/400	APPN
AS/400	AS/400 PerformanceEdge
C/400	COBOL/400
Data Propagator	DATABASE 2 OS/400
DB2/400	Distributed Relational Database Architecture
DRDA	IBM
ILE	ImagePlus
OS/400	RISC System/6000
RPG/400	RS/6000
S/390	SNAP/SHOT
SQL/400	SystemView
VTAM	

The following terms are trademarks of other companies:

386,486	Intel Corporation
HP-UX	Hewlett-Packard Company
NetWare	Novell, Inc
Macintosh	Apple Computer, Inc.
Microsoft, Windows	Microsoft Corporation
SCO Open Desktop	SCO UNIX 386
SunOS/Solaris	Sun Microsystems, Inc.
1-2-3, Lotus, Freelance, Freelance Graphics	Lotus Development Corporation.
BAPCo	Business Application Performance
UNIX	licensed exclusively through X/Open Company Limited

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to discuss AS/400 performance management, minimize the impact of performance problems, and expedite the resolution of performance problems on the AS/400. It also collects a large amount of the performance information scattered in several sources of information and presents use of this information in an ordered manner. Where there is a previously developed technique for performance management and performance problem analysis, portions of this document can be integrated into those existing procedures. Key objectives of this publication include:

- Make AS/400 customers more self-sufficient
- Improve productivity of IBM SE and business partner
- Increase IBM SE′s added value
- Have a mechanism in place for early and easy planning of upgrades
- Avoid many critical customer situations for performance reasons

A wide spectrum of issues related to AS/400 performance are addressed in this redbook. However, some AS/400 facilities are beyond the current scope of this publication.

This document contains performance recommendations and tips on OS/400, LAN Server/400, AS/400 Client Access/400 (PC Support/400), Structured Query Language (SQL), the Integrated Language Environment (ILE), RPG IV/400, COBOL/400, ILE C/400, ADSTAR Distributed Storage Manager/400 (ADSM/400), LAN Resouce Extension and Services/400 (LANRES/400), ImagePlus Workfolder Application Facility/400 (WAF/400), Ultimedia Services/400, DataPropagator/400, and OptiConnect/400. **However, a thorough treatment of each of these subjects is beyond the scope of this redbook**. References are made to other documentation for additional details.

UNIX capabilities, available with the AS/400 Common Programming API (CPA) Toolkit, an orderable feature of OS/400, enables new AS/400 applications to be written using UNIX-based constructs with ILE C/400. The late availability (March 1995) of the CPA/400 Toolkit prevented including UNIX-based performance information in this redbook. The "Related Publications" section of this manual lists additional information on AS/400 performance topics.

What Is New?

This document is a major update of the previous edition, which included Version 2 Release 2 level support. It now includes Version 3 Release 1 level performance-related information.

It is not clear at this time if this redbook will be updated when the Advanced Series PowerPC-based technology systems hardware and software become available or if a completely new redbook will be published exclusively for these new systems.

Key new information provided in this edition includes:

- Updated CPU utilizations based on latest processor models as listed in the appendixes

- BEST/1 updates based on new disk and processor models and features
- New Work Management page fault guidelines
- Additional communications performance-related information
- Summary of V3R1 Database and SQL performance improvements, and query optimization information
- Additional security performance considerations
- New Performance Tools/400 journaling information on the component report
- New Client Access/400 server implementation and related performance considerations
- New LAN Server/400 performance information
- New IBM jobs running under OS/400
- New IBM subsystem monitor routing entry information for specifying storage pool and run time priority for IBM-supplied applications.
- Summary of ILE RPG and C performance-related information
- Summary of ILE-based performance tool information
- Summary of performance tips for ADSM/400, LANRES/400, ImagePlus WAF/400, Ultimedia System Facilities, DataPropagator/400, and OptiConnect/400
- Additional detail on Set Object Access and Expert Cache support
- Expanded documentation of the Performance Management/400 service offering support and its associated history commands

The Version 3 Release 1 licensed program AS/400 Performance Tools, 5763-PT1, is used as the basis for most of the performance data analysis discussed in this document. Throughout the document other tools may be referenced, but whenever the phrase ″Performance Tools″ is used we are referring to 5763-PT1 with the Manager (full-function) feature.

Audiences

This publication is intended for use by IBM system engineers (SEs), business partners and IBM customers. It is presumed that readers are very familiar with the capabilities of the Performance Tools licensed program and other system commands that are useful in collecting and analyzing performance data. The intent of this document is to increase the productivity of those already knowledgeable of at least some AS/400 performance considerations and tools by presenting an ordered structure for setting up performance management.

Although this publication is available directly to customers, it does contain references to internal IBM documentation and programs. Customer access to, and use of, these internal tools is only through IBM system engineers and authorized business partners. It is at the SE or business partner discretion to determine when a customer can benefit from use of any of these tools. The customer is cautioned that these tools are used ″as is,″ and there is no commitment to resolve problems found in these tools.

However, feedback from the use of these tools is encouraged either via the reader′s comments form of this publication or via the IBM VM tools disk support used to retrieve some of these tools. This feedback must be submitted by the

IBM SE or business partner and will be utilized as part of the prioritization process for future formal product enhancements.

Other Offerings

In the United States, IBM offers a billable ″performance examination,″ which includes a formal presentation and report. This offering is conducted by IBM or IBM authorized business partners for those customers who choose not to develop the necessary expertise within their employ. Other vendors and other countries may also provide equivalent services.

IBM US also has announced a comprehensive set of AS/400 technical service offerings called AS/400 Support Family. With the options available in the Support Family menu of services, a customized total support package can be tailored to your specific requirements. As your needs and requirements change, you can alter your service selections.

One of the services is PERFORMANCE MANAGEMENT/400 (PM/400). PM/400 provides automated performance data collection, summarization and trend analysis to assist you in monitoring system performance and efficiency. The summarized data is transmitted to a central site where the data is analyzed and reports showing trends in system resource usage are produced. These reports are made available according to contractual agreements - monthly, quarterly, and annually.

PM/400 is a world-wide service offering, but is not available in all countries.

IBM offers the AS/400 Performance Analysis and Capacity Planning course (Course Code S0267). This class provides students with in-depth training on AS/400 performance and use of performance tools.

IBM Rochester also offers the AS/400 Performance Workshop course (Course Code S6040). This class allows students to work on their own performance data and learn how to find solutions for their own systems.

Non-US countries also offer other performance-based services. Contact the local AS/400 representative to determine those services available within each country.

This document does not obviate the purpose or value of a formal performance examination service or a performance class. The service is primarily intended for customers who do not have AS/400 performance analysis expertise. The class provides appropriate training for those who choose to develop their own performance skills.

How this Document is Organized

The document is organized as follows:

- Chapter 1, "Introduction to Performance Management" explains performance management, what benefits it has, and who should use it. It also shows you the structure of performance management and how it can be implemented effectively.
- Chapter 2, "Performance Requirements and Objectives" discusses how to establish performance objectives and also explains what can be realized and accomplished on AS/400 and what cannot be expected.
- Chapter 3, "Factors Affecting Performance" draws together the major elements contributing to response time and discusses the key elements that affect performance on the AS/400.
- Chapter 4, "Performance Management Methodology" shows an overall picture of the performance management structure. There is a flowchart giving a general overview of the methodology.
- Chapter 5, "Performance Management and Review" explains how to set up the collection of performance data, what kind of data should be collected, and how to automate the process. It also presents an overview of the options available to review the data.
- Chapter 6, "Performance Trend Analysis" shows an approach to trend analysis which would assist in timely identification of potentially out-of-line conditions and trends.
- Chapter 7, "Performance Problem Analysis" presents a structured approach to a subject widely known to SEs and business partners, and gives guidelines where to find pertinent information in performance reports, charts, and other data.
- Chapter 8, "Additional Performance Tools" contains a short description of other tools and facilities available for the study of AS/400 performance and problem identification.
- Chapter 9, "System Performance Tuning Tips" contains tips and techniques for improving performance of the AS/400 operating system.
- Chapter 10, "Design and Coding Tips" includes application design and coding considerations for DB2/400, query and SQL/400, workstation programming, RPG/400, COBOL/400, C/400, and some client server applications.
- Appendix A, "Guidelines for Interpreting Performance Data" contains important tables on guidelines of how to use resource utilization data and transaction guidelines, and specifies thresholds for exceptions. Note that some of the guideline numbers may be different from previously available charts. The numbers used in this appendix are based on the experience of the authors of this publication.
- Appendix B, "Field Descriptions and Sample Performance Reports" presents samples of the various reports generated by the Performance Tools for the convenience of the users of this document. A selection of less familiar, but important, fields in the reports are also discussed.
- Appendix C, "IBM Internal Use Only Tools/Documents" contains a description of IBM internal tools and facilities available for the study of

AS/400 performance and problem identification; these facilities are available on an ″as-is″ basis.

- Appendix D, “Performance Tools/400 Transaction Boundary Overview” contains a description of the steps within an interactive transaction boundary. The Performance Tools/400 Transaction Report - Transition Details indicates some of these steps in its output.
- Appendix E, “OS/400 Expert Cache and Set Object Access Overview” describes details of OS/400 Set Object Access (SETOBJACC) command and Expert Cache support. Standard AS/400 user documentation provides very high level documentation of these OS/400 ″object caching″ facilities. This section provides some details for determining when to select one of these options and how they work for you to get maximum performance improvement.
- Appendix F, “IBM Communication Jobs and Subsystem Routing Entries” provides the user with compare values to control storage pool assignment and run time priority of important IBM-provided software. This IBM-provided software can then be managed just as standard application programs and job are currently done.
- Appendix G, “PC Support/400 Shared Folder Type 2 Performance Query” provides an overview of V2R3 PC Support/400 shared folder type 2 implementation and sample Performance Monitor database file queries.

  These queries may be used to measure and capacity plan shared folder type 2 workloads on releases V2R2 and V2R3.

  Note that V3R1 implementation of Client Access/400 for equivalent function is significantly different.
- Appendix H, “Sample X.25 Queries for Network Congestion” provides sample queries of X.25 related Performance Monitor data to evaluate if the AS/400 or the network are the cause of X.25 network congestion - packet frames are rejected by either the AS/400 or the network.

  The Performance Tools/400 reports X.25 statistics but does not include congestion information.

Related Publications

The following publications are considered particularly suitable for additional information on AS/400 performance topics.

Prerequisite Publications

- *AS/400 Work Management Guide Version 3*, SC41-3306
- *AS/400 Performance Tools/400, Version 3*, SC41-3340
- *AS/400 DB2/400 Database Programming Version 3*, SC41-3701
- *AS/400 DB2/400 SQL Programming Version 3*, SC41-3611
- *AS/400 DB2/400 SQL Reference Version 3*, SC41-3612
- *OS/400 Server Concepts and Administration Version 3*, SC41-3740

The following are articles contained within the identified issue of *NEWS 3X/400*, a non-IBM magazine issued monthly which contains articles contributed by both IBM and non-IBM authors. While all 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 AS/400 and System/36 performance.

- *″System Performance: Comm Line Time,″ NEWS 3X/400*, March 1991
- *″System Performance: CPU Time,″ NEWS 3X/400*, April 1991
- *″System Performance: Disk Access Time,″ NEWS 3X/400*, May 1991
- *″System Performance: Controlling Demand on Memory,″ NEWS 3X/400*, June 1991
- *″Object Contention and Performance Analysis,″ NEWS 3X/400*, July 1991
- *″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 441 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
- *″Defining System/38 Performance Requirements,″ NEWS/34-38*, May 1988
- *″Measuring System/38 Performance,″ NEWS/34-38*, June 1988
- *″Improving System/38 Performance,″ NEWS/34-38*, August 1988

The *AS/400 Magazine* produced by the US Competency Center in Rochester, MN, USA, 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 Programming: Version 3 Release 1 Performance Capabilities Reference*, ZC41-8166

  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.

  This manual also lists OS/400 and many licensed program module names. The *System Support Diagnostic Aids (Volume 1)*, LY44-0597 (available to IBM licensed customers only), lists Licensed Internal Code (LIC) tasks.

  To ensure you have access to the latest update to this manual, we recommend using the following VM command to obtain a copy:

  ```
  REQUEST V3R1 FROM FIELDSIT AT RCHVMW2 (your name
  ```

International Technical Support Organization Publications

The following ITSO publications provide important performance information on specific AS/400 facilities:

- *PC Support/400 Implementation and Performance*, GG24-3636
- *Structured Query Language/400: A Guide for Implementation*, GG24-3321
- *LAN Server/400: A Guide to Using the AS/400 as a File Server*, GG24-4378
- *Client Access/400 for Microsoft Windows 3.1*, GG24-4429
- *An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, Communications and PC Connectivity*, GG24-4200

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

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

To get a catalog of ITSO technical publications (known as “redbooks”), VNET users may type:

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

How to Order ITSO Technical Publications

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

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain books on a variety of products.

Below is a list of ITSO publications that are currently available which relate to the AS/400.

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

- *System/36 to AS/400 System Migration*, GG24-3249-01
- *System/36 to AS/400 Application Migration*, GG24-3250-01
- *AS/400: System/38 Application Migration to AS/400*, GG24-3251-00
- *AS/400 Communication Migration*, GG24-3253--00
- *AS/400 Office in a DIA/SNADS Network*, GG24-3268-00
- *Converting S/36 Environment Application to Native*, GG24-3304-01
- *AS/400 Communications Problem Determination*, GG24-3305-00
- *SQL/400: A Guide for Implementation OS/400 V2R2*, GG24-3321-03
- *AS/400 - S/370 Connectivity*, GG24-3336-00
- *AS/400, S/38 and PS/2 as T2.1 Nodes in a Subarea Network*, GG24-3420-00
- *Writing SAA Applications for AS/400*, GG24-3438-00
- *IBM AS/400 TCP/IP Operation and Configuration*, GG24-3442-02
- *IBM AS/400 in Large Networks: A Case Study*, GG24-3447-00
- *AS/400 Communications Definitions Examples*, GG24-3449-00
- *AS/400 Object Distribution Facility and SNA RSCS PROFS*, GG24-3479-00
- *IBM AS/400 ISDN Connectivity*, GG24-3517-00
- *OfficeVision/400 and AS/400 Query Applications in a Multilingual Environment*, GG24-3579--00
- *Managing Multiple AS/400s in a Peer Network*, GG24-3614-02
- *OfficeVision/400 in a DIA/SNADS Network*, GG24-3625-00
- *AS/400 Audit and Security Enhancements in OS/400*, GG24-3639-00
- *WAF/400 5363 Optical Subsystem Configuration and Installation*, GG24-3680-00
- *OfficeVision/400 Printing*, GG24-3697-00
- *AS/400 Printing II*, GG24-3704-00
- *AS/400 APPN with PS/2 APPN, 3174 APPN, 5394 and Subareas*, GG24-3717-00
- *AS/400 CPI Communications Selected Topics*, GG24-3722-00
- *AS/400 Performance Management V2R2*, GG24-3723-01
- *Multimedia Examples with the AS/400 Using AVC*, GG24-3743-00
- *Getting Started with AS/400 OSI*, GG24-3758-00
- *AS/400 Communication Definition Examples Volume 2*, GG24-3763-00
- *Installation Considerations for National Language*, GG24-3790-00
- *Artificial Intelligence and AS/400: Neural Networks and Knowledge Based Systems*, GG24-3793-00
- *Facsimile Support/400 Implementation*, GG24-3797-00
- *Application Development on the AS/400*, GG24-3806-00

- *PC Support/400 Asynchronous and SDLC Configuration Examples*, GG24-3808-00
- *5494 & OS/2 ES: Connecting Remote User Groups*, GG24-3828-00
- *AS/400 Automation Using NetView and SNA MS Transport*, GG24-3841-00
- *DOS PCS/400 in OS/2 V2 Virtual DOS Machine*, GG24-3856-00
- *WAF/400 Administration and User Examples*, GG24-3866-00
- *OfficeVision/400 Application Enabler*, GG24-3868-00
- *Cooperative Processing and GUI in an AS/400 Environment*, GG24-3877-00
- *OfficeVision/400 Application Programming Interfaces V2R2*, GG24-3885-00
- *OfficeVision/400 Integration with CallPath/400 and Fax Support*, GG24-3896-00
- *AS/400 Performance Capacity Planning V2R2*, GG24-3908-00
- *AS/400 System Availability and Recovery for V2R2*, GG24-3912-00
- *AS/400 Network Routing Facility*, GG24-3918-00
- *AD/CYCLE Code/400, ADM/400 and ADS/400*, GG24-3928-00
- *OfficeVision/400 V2 Technical Tips and Techniques*, GG24-3937-00
- *CICS/400 Migration from Mainframe CICS*, GG24-4006-00
- *Using DOS PC Support/400 with Novell NetWare 3.11 and NetWare for SAA 1.3*, GG24-4013-00
- *Ultimedia Video Delivery System/400*, GG24-4020-00
- *AS/400 Client Series - Products and Positioning*, GG24-4027-01
- *IBM AS/400 Printing III*, GG24-4028-00
- *Performance Benchmarking for the AS/400*, GG24-4030-00
- *AS/400 and RISC System/6000 Connectivity*, GG24-4039-00
- *Using V2R3 DOS and OS/2 PC Support/400 under OS/2 2.1*, GG24-4070-01
- *Apple Macintosh and the AS/400*, GG24-4071-00
- *OfficeVision/400 Application Enabler Version 2 Release 3*, GG24-4072-00
- *The IBM AS/400 as a TCP/IP Network File Server*, GG24-4092-00
- *ENVY/400 Hints and Tips*, GG24-4094-00
- *Introduction to ENVY/400*, GG24-4126-00
- *Managing Operations on AS/400s with IBM SAA SystemView OMEGAMON Services/400*, GG24-4136-00
- *AS/400 Integrated Language Environment*, GG24-4148-00
- *CICS/400 V2R3 Task Book*, GG24-4182-00
- *AS/400 V2R3 Software Life Cycle Mgmt with ADM/400*, GG24-4187-00
- *An Implementation Guide for AS/400 Security and Auditing including C2, Cryptography, Communications and PC Connectivity*, GG24-4200-00
- *IBM AS/400 APPN Problem Management*, GG24-4222-00
- *V2R3 PC Support/400 and Microsoft Windows 3.1 Advanced Topics*, GG24-4253-00
- *OfficeVision/400: Printer Setup in an OfficeVision Environment*, GG24-4283-00
- *AS/400 Client Series Handbook*, GG24-4285-00
- *Backup Recovery and Media Services/400 Implementation Tips and Techniques*, GG24-4300-00
- *IBM Current-OV/400 Workgroup Program V1 R1 Modification 0 Refresh 1*, GG24-4377-00
- *Client Access/400 Planning Guide*, GG24-4422-00

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 130NC

This item summarizes performance-related PTFs and is continually updated. This item should be reviewed monthly for PTFs applicable to various customer environments.

- Item 226NC

  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, etc.

- Item 7WBDK

  This item summarizes interactive and batch performance expectations for the AS/400 Server Model series of systems.

- *Performance Capabilities Reference* manual, ZC41-8166

  This is a VM listing hardcopy of most of the items referenced by 226NC. New versions are made available at key announcement and general availability dates. Because of publications schedules, an update may be made to the corresponding item referenced by 226NC, but a hardcopy update may not be available until a later date.

  Item 226NC describes how IBM representatives and authorized IBM business partners can obtain this hardcopy manual.

- Item 8WFWM

  This item is a summary of AS/400 server model specific performance information. It serves as a reference to other server model performance-related HONE items.

  Many of the items referenced in item 8WMWF are also included in the *Performance Capabilities Reference* manual.

Acknowledgments

This project was designed and managed by:

Jim Cook, International Technical Support Organization (ITSO) Rochester

The authors of this document are:

Lou Antoniolli, IBM Asia Pacific, Melbourne, Australia
Joy Perez,IBM Philippines, Manila
Neil Willis, IBM Australia, Sydney,
Garry Wood, IBM New Zealand, Wellington

The advisor of the second (previous) edition of this document was:

Maggie Haase, IBM Rochester, Minnesota

The authors of the second edition of this document were:

Alvaro Alliende, previously of IBM, Chile
Zainal Hanafi, IBM Mid-Range Support, Indonesia
Kathia Espinosa Mascarenhas, IBM Field Support Center, Brazil
Petri Nuutinen, IBM Field Support Center, Finland
Lloyd Perera, IBM Sydney, Australia

The advisor of the first edition of this document was:

Jim Cook, ITSO Rochester

The authors of the first edition of this document were:

Gustavo Leiva, IBM Customer Center, Chile
Neville Hilton-Taylor, ISM South Africa
Trond Russ, formerly of IBM Norway
Gottfried Schimunek, IBM Field Support Center, Germany

Communications performance information was developed by the the following residents:

Alfred Tan, IBM Singapore
Fernando Zulieani, IBM Brazil
Gary Zuccala, IBM Australia
Linda Allen, IBM US Competency Center
Wally Schwane, Rochester Development

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

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

International Technical Support Organization, Rochester Center including:

- Marcela Adan
- Wilfried Blankertz
- Michele Chilanti
- Amit Dave
- Nick Hutt
- Ian Jarman
- Gary Mullen-Schultz
- Rob Stagg
- Klaus Subtil
- Craig Tamlin
- Ted Zonderland

Key Development Lab and US Competency Center contributors and reviewers include:

Bindhu Aravapalli, IBM Business Partner
Dick Bains, Rocheste Development
Toby Barksdale, Rochester As/400 Competency Center
Pam Brady, Rochester Development
John Bazey, Rochester Development
Jim Coon, Rochester Development
Larry Cravens, Rochester AS/400 Competency Center
Paul Day, Rochester Development

Mike Denney, Rochester Development
Dave Ellefson, Rochester Development
Dan Frissora, Rochester Development
Ben McIntosh, Rochester AS/400 Competency Center
Chuck Stupca, Rochester Development
Tom Strong, Atlanta AS/400 Education
Kurt Thompson, Rochester AS/400 Competency Center
Rick Turner, Rick Turner Associates
Pam Wright, Rochester Development

For V3R1 updates special thanks go to:

- Bruce Adelsman, Randy Egan, Dave Weber for Query and SQL performance tips
- John Bazey for Client Access/400 work management
- Entire AS/400 Performance Evaluation departments, especially Charlie Farrell, Kim Greene, Dave Johnson, Dale Weber, Joe Parks, Dave Hoffman, Rick Peterson, Brad Nelson, Richard Shannon, and Wally Schwane
- Tom McKinley and Rick Turner for object caching and expert cache information
- Ian Jarman for Client Access/400 host server information
- John Bazey, Darla Klepel, Jan Dunston (IBM South Africa), Gary Mullen-Schultz, Bob Maatta, and Nick Hutt for V3R1 Client Access/400 information
- Rob Stagg for LAN Server/400 information
- Marty Keech and Harold Romo for APPC tips and IBM Routing Entry Compare Values

Chapter 1. Introduction to Performance Management

1.1 What Is 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 (whatever that means to the customer).

The IBM SystemView* announcement document (290-539), dated September 1990, stated performance management covers capacity planning, collection of performance data, and tuning of resources.

1.2 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 very 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 could 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 very necessary in determining trends and possible trouble areas. For example, it is very 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 will identify 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) could lead to 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.

1.3 Performance Management Methodology

1.3.1 Components of the Methodology

This document identifies the following components of performance management:

1. Setting performance objectives
2. Performance measurement
3. Performance data analysis
4. Performance trend analysis
5. 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.

We decided to use the Performance Tools/400 Licensed Program (5763-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 5763-PT1.

AS/400 query products can be used to process the Performance Monitor data in ways that are not directly available via 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 which 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.

1.4 Benefits of Using the Performance Management Methodology

1.4.1 Benefits for the AS/400 Customer

The following are some of the things that the AS/400 customers can benefits 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.

1.4.2 Benefits for IBM and Business Partners

The following are 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.

1.5 Prerequisites

1.5.1 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.

1.5.2 Other Publications, References

We did not intend to reproduce formal publications containing AS/400 performance information such as the *Performance Tools/400 Guide*, SC41-3340, 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 in this document, much information 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 the "Related Publications" section of this document for additional AS/400 performance documentation.

1.5.3 IBM Performance Tools

For Version 3 Release 1, the Performance Tools/400, 5763-PT1, contain support for the new hardware, new page faults per second guidelines, journaling entry statistics, and include DDI (Distributed Data Interface) and Frame Relay (FRLY) data in the communications-related reports.

From a licensed program ordering viewpoint, the Version 3 Release 1 Performance Tools/400 have been repackaged and renamed from previous releases, such as V2R3 5738-PT1. For V3R1 the full function Performance Tools/400 are ordered as a ″base function″ and a ″manager feature″ (H1). For V3R1 the subset function is ordered by selecting the ″base function″ and the ″agent feature″ (H2).

The Agent Feature function set is similar to that available in previous releases with the Performance Tools Subset/400 (5798-RYP). The Agent feature provides a major enhancement compared to the Performance Tools Subset/400 by supporting 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.

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 a system job (QPFRCOL) be started that looks at the performance collection start values and starts and ends the Performance Monitor accordingly.

Either via 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 via the Advisor function (ANZPFRDTA command)
- creation of a BEST/1 model (CRTBESTMDL command)
- analysis of a BEST/1 model (ANZBESTMDL command)

The *Performance Tools/400 Guide for Version 2*, SC41-8084-02, *Performance Tools/400 for Version 3*, SC41-3340, *BEST/1 Capacity Planning Tool Guide for Version 2*, SC41-0139, and *BEST/1 Capacity Planning Tool for Version 3*, SC41-3341, describe the automation capabilities and the use of the performance tools.

Chapter 8, "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 Only″ or ″as is″ tools available from the 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 can be found in 8.5, "Performance Management/400 (PM/400)" on page 119.

1.6 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 this 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 via 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

2.1 Defining AS/400 Performance 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 via 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 screen 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′s important to keep the different processing requirements in mind. Often the analyst′s perception of adequate performance will 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.

2.2 Criteria for Interactive Jobs

Considering interactive jobs, one definition of good performance is that response time is good enough to ensure that customers don′t 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/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 all objectives should be based on average values. Refer to Chapter 3, “Factors Affecting Performance” on page 11 and 3.1, “Interactive Response Time Components” on page 11 for more details.

2.3 Non-interactive (Batch) Criteria

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

Non-interactive criteria could be 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 spool file entry appearing on an output queue. A ″non-interactive transaction″ could include job run time, time to complete printing a spool 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(s). In the ADSTAR Distributed Storage Manager/400 (ADSM/400) environment it could include the time taken to ″backup″ or ″restore″ files and directories between the AS/400 and one or more client workstations.

In many cases knowing the number of records processed per unit of time or 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 what defines the unit of measurement by which they judge performance to be acceptable or non-acceptable.

2.4 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.

2.5 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 subsecond response time on any current models of the AS/400. And using full screens of data to ten or more workstations on a 4800 bps (bits per second) remote communications line will not deliver subsecond response time, regardless of CPU instruction speed. The following examples illustrate what can and cannot be expected on the AS/400:

- Running through 500 statements of RPG or COBOL code and twenty or less database I/Os per transaction on a mid-sized AS/400 will probably give you very reasonable response times and very 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 will rarely result in consistent subsecond response times.

  The Performance Tools reports can determine the average of actual disk reads and writes (disk physical I/O) per transaction. Use the Disk Physical I/O per Transaction Guidelines in Table 72 on page 352 as a guide.
- Filling a subfile with a thousand records before displaying the first screen will 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 will result 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 will usually result in acceptable performance. In many Personal Computer (PC) workstation LAN environments, using any of 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, achieve response time 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 screen data is much less than the data including many database records.
- When two-phase commitment is used via 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 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, then that job may ″use more of the CPU″ and thus 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 on-line 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
  - Using 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 9, “System Performance Tuning Tips” and Chapter 10, “Design and Coding Tips” for more a wide range of performance tips.

Chapter 3. Factors Affecting Performance

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.

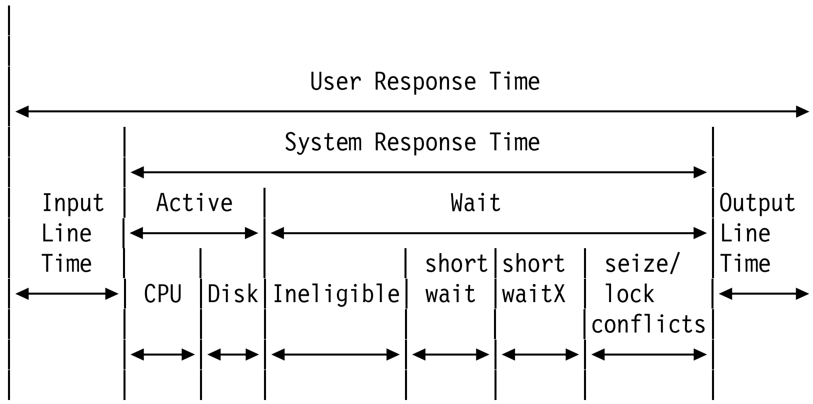


Figure 1. Components of Interactive Response Time

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 will be the total of many components:

- There will be 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 will begin.
  - 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 queuing)
    - Disk I/O time (including queuing)
  - There may also be periods of inactivity during which the transaction would be waiting in a variety of states, including:
    - Ineligible activity time (Excs ACTM)
    - Short waits
    - Short waits - extended
    - Object or record seize/lock conflict
- Finally, there will be a transmission delay in the response reaching the user.

Note: If you are running pass-through or 3270 emulation, the output line time will also include 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, etc.), the processor speed and any other work being done concurrently on the remote system/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*, SC41-3340, 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/or ″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.

Beginning with Version 2 Release 2, 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 (DWS) attached to the IBM 5494 Controller. Apart from this, **neither Performance Monitor data nor Performance Tools response time output include communication line time**. Overall communication line information is collected by the Performance Monitor and the collected data can be output by the Performance Tools or user-written queries of the QAPMxxx database files as described in the *Work Management Guide*. Refer to Chapter 9, "System Performance Tuning Tips," 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. Without it, you cannot define the target values you need to achieve.

  What is the contribution of each component (Figure 1 on page 11) to performance? Determine this for both transactions that are, and are not performing as they should.

- Collect comparative data before and after changes are made.

  This will enable you to evaluate the effect of the changes you have made.

- How do you identify which component is causing problems? It could be that one or more is out of range and need 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 lack of 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, then 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

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 of the request is called the **service time.**

Note: A single server can service only one request at a time. Multiple requests will 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 or complete a cash withdrawal at the ATM or 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 at random intervals
- Requests for the resources are not all the same.

As the utilization of a server increases (more work for the server), queuing can account for much longer work (or request) completion. In an interactive transaction, this could be considered the cause of long response times. The Queuing Multiplier (QM) is a measure of queuing. Table 69 on page 348 contains a table showing the approximated 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 3. A QM of 3 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) would take a minimum of .5 seconds to use the CPU (server response time = QM * 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, 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, need to be factored into expectations. These Exceptional Wait Times (waiting for record or object locks, waiting for communication line data transmission, etc.) 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 which assumes a CPU QM of 4 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 4 will generate less accurate results.

The Performance Tools Transaction Report - Job Summary lists the CPU Queuing Multiplier determined 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 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 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 react the same. There are different recommended maximum values for the different resources, such as CPU, disk, memory, controller, remote line, IOPs, etc.

The *AS/400 Performance Tools Guide* provides more queuing information.

The following graph shows a simplified queuing formula, and a curve derived from it highlighting the effect of increasing utilization on the Queuing Multiplier for a single server:

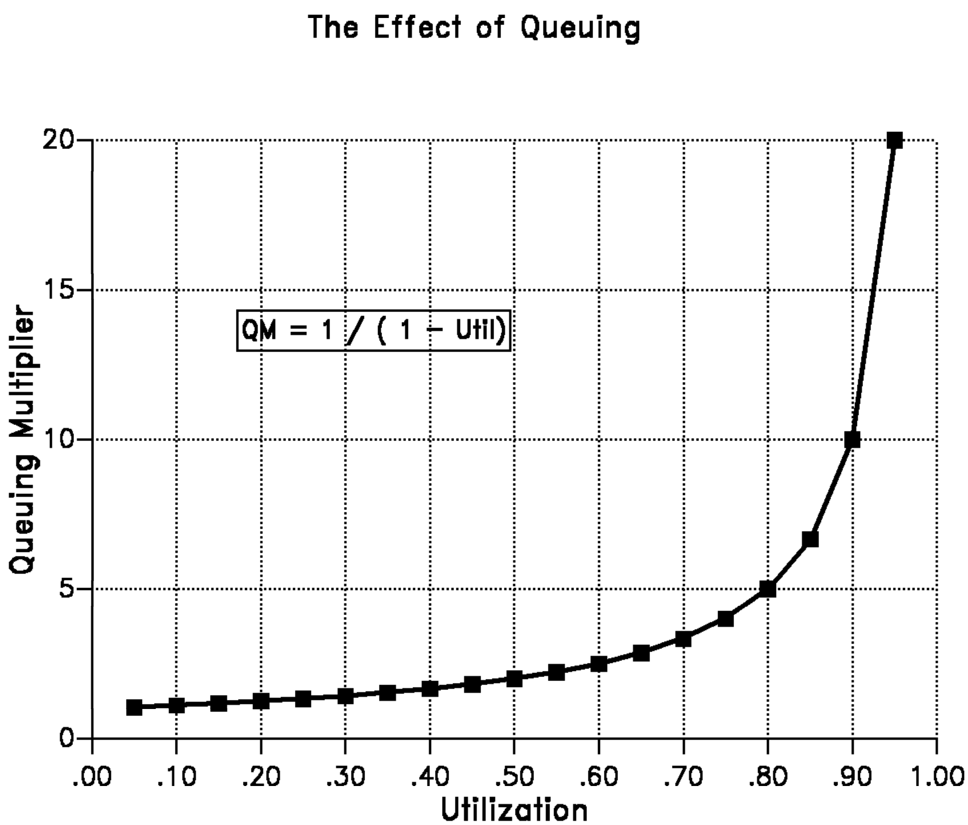


Figure 2. Queing Effect: Simple Equation for a Single Server and Random Arrivals

3.3 Additional Queuing Considerations

3.3.1 Multiple Servers

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

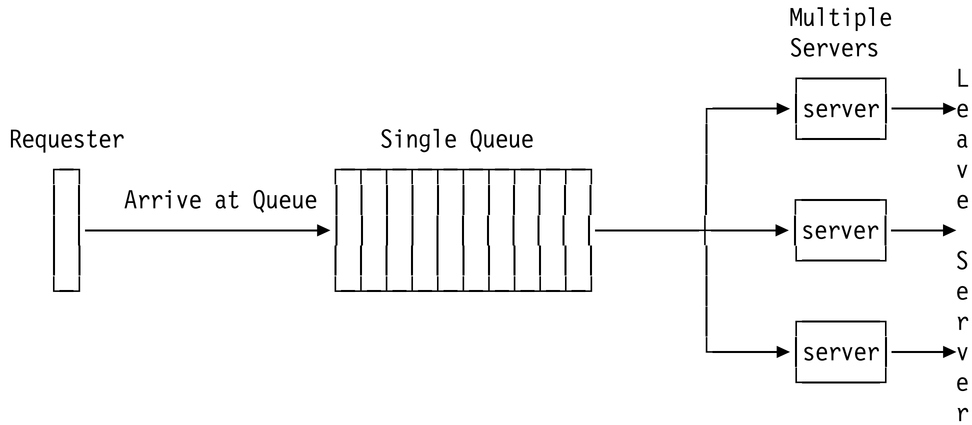


Figure 3. Single Queue with Multiple Servers

Under these conditions the Queuing Multiplier equation can be represented by:

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

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

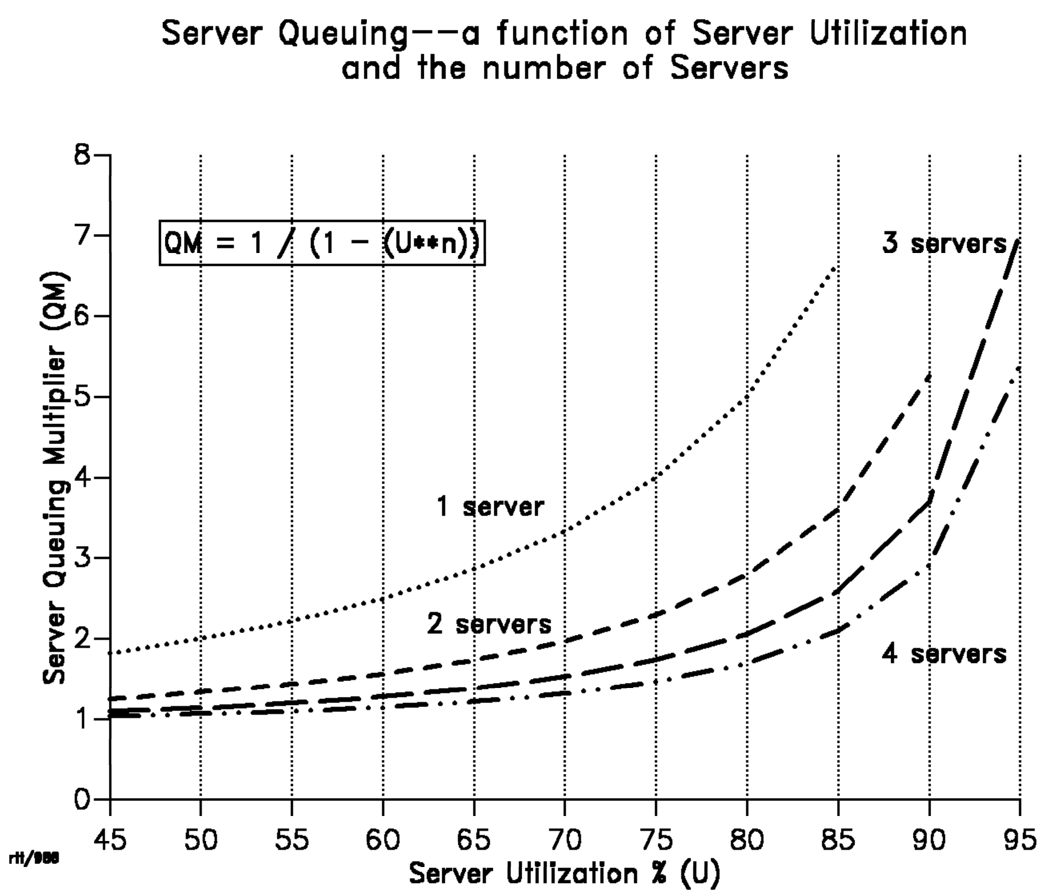


Figure 4. Queuing Effect: Simple Queuing Equation for a Multiple Server

For a given Queuing Multiplier (QM), the rate at which QM changes with increasing utilization (slope at that point in the QM vs Utilization graph) is higher as the number of servers increase. This means that for a given QM value, multiple-server systems are more sensitive to increases in utilization than single-server systems.

The objective of setting resource utilization thresholds is to minimize queuing and provide a stable environment for consistent response time. Thus, in the various N-way processor environments, the rate of change of the slope is used to determine optimum, average processor utilization values for the multiple-processor systems.

Refer to Table 24 on page 323 in Appendix A on N-way processor systems.

3.3.2 Other Factors that Affect Queuing

Two additional factors also play an important part in Queuing Multiplier 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
```

3.4 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 too long for the CPU, lose 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 Licensed Internal Code (LIC) tasks in the system (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 Manual* LY44-3900 (available to IBM-licensed customers only).

On many AS/400s, LIC tasks may normally use up to 6% to 12% of the total CPU capacity, while OS/400 subsystem and monitor jobs normally use about 1% to 3%.

High CPU time in LIC 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 in this guideline; they have high I/O rates due to the nature of the work they perform.

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

3.4.1 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 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.

Example

Assuming CPU% = 50% and a transaction takes 0.5 seconds of CPU time, the time to use the CPU would be 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
```

3.4.2 CPU Guidelines

Stable CPU service 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, then all work at priorities 00-19 are also included) is below:

- 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 1-way processor at 70% utilization, an increase in utilization by 5% may still result in acceptable response time. However, for a 3-way processor, a 5% increase in utilization may result in unacceptable response time.

3.4.3 CPU Relative Internal Processor (RIP) Performance

The AS/400 family has a tradition of rating the relative processor speed among models with a *relative internal processor (RIP)* performance value, where a value of ″1″ represents the original and slowest AS/400, the model B10. The RIPs 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 RIP value represents testing results with an interactive internal IBM benchmark - RAMP-C. This rating remains through Version 3 Release 1 and has proven a good guideline for capacity planning of system resources. For the AS/400 server model series of systems a different internal IBM benchmark involving remote SQL and other client/server access methods was used to establish the non-interactive RIP values.

A RIP may not apply to all application environments as their mixture of machine instructions run by the system could be significantly different than 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 a RIP of 4.7 and an F50 has a RIP of 9.5, roughly 2 times the F35.

In an interactive environment the F50 should deliver generally 2 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 then the batch jobs should achieve the expected run time when comparing the F35 and the F50.

Remember also that the RIP ratings for the multiprocessor models presume multiple jobs active concurrently. Run time expectations for a single batch job should be based on the single processor RIP values, such as the F70 and 320-2050 models rather than the multiprocessor models.

3.4.4 Server Model Series Considerations

The AS/400 Server Models were introduced in 1993. When compared to the ″traditional series models″ (non-server models), these server models provide a significantly different implementation of interactive versus batch performance. These models were introduced to make the AS/400 attractive for customer environments that were primarily client/server oriented without the cost of a full function CPU model. In order to deliver client/server job performance on a significantly lower cost processor, the server models are ″tuned″ specifically for non-interactive work while limiting the performance of interactive workstation-based sessions and the number of concurrently active interactive sessions.

This tuning is why interactive work receives the lower RIP capabilities on the server models, even when no non-interactive work is being done.

On the server models any job, not recorded as an *interactive* job by the Performance Monitor, receives the benefits of the non-interactve tuning. Any jobs not showing ″INT″ on the Work with Active Jobs (WRKACTJOB) command output are considered non-interactive jobs. When viewing performance monitor database files a value of ″I″ in field JBTYPE of database file QAPMJOBS verifies the associated job is considered as an interactive job by the system. In general an ″I″ type job includes a normal 5250 and 3270 workstation job, display station pass-through jobs, PC Support/400 Work Station Function (WSF) jobs, any OS/2 Communications Manager 5250 emulation job, any RUMBA/400 display job, or DHCF/Network Routing Facility/SNA Primary Logical Unit job.

Two sets of RIP values are available on the server series models - interactive and ″non-interactive.″ Table 1 shows an example of a server model and traditional model RIP values.

Table 1. Server - Traditional Model RIP Comparison Example

System Model	Interactive RIP	Non-interactive RIP
Model 30S-2411	3.3	10.9
Model D70	10.3	10.3

As you can read, the 30S-2411 has non-interactive RIP slightly higher than a D70. The 30S-2411 interactive RIP of 3.3 is equivalent to an F10. The traditional D70 has the *same* RIP for both in interactive and non-interactive work. As an additional traditional model comparison, the F50 has a RIP of 9.5 and the 9406-2043 has a non-interactive RIP value of 10.9.

The server models have a Licensed Internal Code implementation that ″favors″ a non-interactive job more than an interactive job. As previously stated, this remains in effect for interactive jobs even if there are no non-interactive jobs active.

Feedback from server model customers indicate they are satisfied with their non-interactive performance and get slightly better interactive performance when comparing a server model and a traditional model with the same interactive RIP value, such as the 30S-2411 and the F10.

Note that all server models place a restriction of a maximum of 7 concurrently interactive workstation sessions as follows:

- Workstation sessions counted in the 7 maximum limit
  - 5250 display devices locally attached via twinaxial connections. This includes 5250 display emulators that are *not using APPC to communicate to the AS/400* and applies to remote attachment of 5250 displays as stated below.
  - 5250 display devices attached via 5251 Model 12, 5294, 5394, and 5494 remote controllers
  - 3270 display devices attached via 3274 and 3174 remote controllers
  - 3270 display sessions routed to the AS/400 via an SNA backbone network, including DHCF, Network Routing Facility (NRF), and SNA Primary Logical Unit Support (SPLS).
- Workstation sessions not counted in the 7 maximum limit
  - 5250 display device sessions implemented with 5250 data stream functions encapsulated within APPC protocol.

    This includes:

    - PC Support/400 Work Station Function (WSF), OS/2 Communications Manager "5250 emulation," and RUMBA/400 sessions via personal computers attached to the AS/400 via twinaxial, SDLC, X.25, and token ring communications.
    - PC Support/400 Work Station Function (WSF), OS/2 Communications Manager "5250 emulation," and RUMBA/400 session via personal computers attached to the AS/400 through the 5394 and 5494. communications.
    - Standard Display Station Pass-Through support.
  - 5250 devices connected via the Wireless adapter
  - ASCII Work Station Controller attached port sharing "5150 devices"
  - TCP/IP TELNET sessions
  - 3270 Emulation
  - Apple workstations attached via the AppleTalk local workstation controller

3.5 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 Licensed Internal Code (LIC) tasks are predetermined. Some run under a user job priority while others run at very high priority - higher than user jobs.

A basic understanding of task dispatching can be helpful when analyzing problems like performance degradation or understanding that the server model series algorithm modifies normal usage of the job priority (RUNPTY) attribute for task dispatching.

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

The high-end multiprocessor (N-way) systems also share a single TDQ. Refer to 3.3.1, "Multiple Servers" on page 15, for a 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
- Cache affinity (multiprocessors only)

Selecting which tasks (OS/400 jobs and LIC tasks) to run in which processor is done through a combination of priority, eligibility, and cache affinity . If all processors for which a task is eligible are busy with a higher priority task, the waiting task is not dispatched.

With the multiprocessor systems cache affinity is used to dispatch tasks to the processor on which they are most likely to have residual data in main storage cache. This means an attempt will be made to load the program into the processor it most recently ran in. The task is dispatched only if the processor with which it has cache affinity is available **unless doing so would result in a processor remaining idle or if an excessive number of higher priority tasks might be skipped.** Once this ″skip threshold″ (implemented internally by the LIC) has been reached, cache affinity is ignored and the task is assigned to any processor for which it is eligible. The skip threshold is adjusted to ensure there are no unassigned processors and no skipped tasks.

Initially, a task has equal affinity to all processors. When it is initially dispatched, a processor selection is based only on priority and eligibility.

Note that in the server model series, the interactive jobs have a rather short internal time slice value compared to the time slice value used for non-interative 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.

3.5.1 AS/400 Multiprocessor Main Storage Interleaving

There are memory card placement considerations that enable *full performance benefit of AS/400s with multiple processors*. 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.

There are a maximum of 6 card slots for memory cards. By using memory cards of the same size and in the first four card positions, the maximum benefit of memory interleaving is achieved. Spreading system memory accesses across all 6 memory cards would not result in significant interleaving improvement over the first 4 card slots.

Using less than 4 card slots and mixed storage sizes will prohibit full interleaving.

You should verify card placement that enables full interleaving before upgrading a 3-way and 4-way multiprocessor system. Depending on the application environment, going from no interleaving to full interleaving could improve performance by more than 15%.

3.6 Memory

Memory is a key resource contributing to overall response time. Its impact cannot be directly measured with any of the available performance tools, nor is it possible to determine the memory usage of a particular job with the performance tools.

3.6.1 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, etc.

When either type of fault occurs, the process must wait for the necessary information to be transferred from disk to main storage. These types of disk operations are called synchronous 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

  Eight pages (512 bytes per page) are read in.

- Non-database object

  The number of pages read in is based on the specific implementation of the object. The default is a single page (512 bytes) but most High Level Language programs are designed to be read in as 8 pages.

Other system functions and individual product implementations frequently override these defaults. 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 441 for additional information.

- Use of Set Object Access (SETOBJACC) commnd

  The SETOBJACC command enables the user to tell the system a specific object to load into main storage with the intent of ″pinning the object″ within the storage pool to achieve main storage access speed.

  See this chapter and Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 441 for additional information.

- Object header ″block transfer″ of 8 to 64 pages to bring in

The implementer of a particular object, such as an ILE C/400 object or an AS/400 data queue determines if system defaults are to be used or a block transfer of from 8 to 64 pages are to be brought in.

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

3.6.2 Memory (Storage) Pools

Main storage is segmented into memory (storage) pools to reduce 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 other concurrent jobs sharing the same main storage pool. The main storage requirement is affected by (a) the program size, (b) whether or not the program used by the job is shared by other jobs, and (c) the amount of temporary storage in use (such as file buffers and program variables). Some memory is used for job associated temporary storage 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 screens are generally used to infer demand for main storage.

Note that the WRKSYSSTS pages per second rate includes pages read into the storage pool, not just pages faulted in. Sometimes *paging rates* and *fault rates* are used interchangeably, but in some cases where pages are read into the storage pool ″ahead of use,″ such as when SEQONLY(*YES) is specified for a database file, the pages read are not because of page fault processing. When a fault occurs, and DB or NDB pages have to be read in, then 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, etc.

After this processing, as pages are being examined, referenced pages become unreferenced pages. Database pages become unreferenced sooner than non-database pages.

If changed pages are detected during the selection process, the pages are given to the system page out tasks 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.

Note also, 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 via the routing entries for a specific subsystem. The Add Routing Entry (ADDRTGE) and Change Routing Entry (CHGRTGE) commands include compare values that can be used to route different jobs into different storage pools. Knowledge of the routing entries used for V3R1 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 10, “Design and Coding Tips,” for additional work management information on these client/server applications.

3.6.3 SETOBJACC

V2R2 OS/400 introduced the capability to load specific objects into a specific shared or user (non-shared) 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 will stay 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 another new in V2R2 commmand Clear Pool (CLRPOOL)).

If possible, the CLRPOOL-SETOBJACC process should be performed when the system work has generally been quiesced. This ensures the pool is cleared of all objects and only the designated SETOBJACC object is placed 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 will load it into the pool and not purge the data thereafter. This provides very 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(s) or program object(s) of an application are known, the size of the object(s) 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.

3.6.4 Expert Cache

Expert Cache became available with OS/400 V2R3 and enhances the AS/400 single-level storage use of main memory as a cache. Like SETOBACC it overrides normal storage management algorithms discussed in 3.6.2, “Memory (Storage) Pools” on page 24. Expert Cache causes frequently accessed data to remain in storage for long periods of time. Unlike, SETOBJACC which requires the user to know which objects should be specifically placed into storage, Expert Cache gives System Storage Management and Database Management the responsibility for *determining and managing portions of objects (usually database files)* that should remain in main memory longer than under normal algorithms.

Expert Cache is enabled by the user for a *shared storage pool only* and applies to all objects in that storage pool. As the application environment changes over time, Expert Cache monitors this and will change the objects and portion of an object that are ″pinned″ in main storage.

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

Prior to V2R3 the I/O Cache PRPQ (no longer available with V2R3 and later) gave the user automated tuning similar to Expert Cache but was functionally very different. The PRPQ was **job-oriented** and Expert Cache is **storage pool-oriented**.

3.6.5 Working Set Size

Working Set Size is defined as the amount of main storage necessary to run a program within a job and achieve 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. Gradually increase storage pool size until page faulting is zero. This is more efficient than starting with a very large pool and gradually reducing the pool size until page faulting is observed.

Working set size applies to both IBM software and customer applications. Note that starting in V2R3, the OS/400 job program code and variables storage are managed differently from previous releases. For example, the Program Automatic Storage Area (PASA) exists in name only. Variables are managed in different work spaces and UNIX-like ″heap terminology″ is introduced. For more information on this new job structure refer to “DSPACCGRP and ANZACCGRP” on page 107.

V3R1 contains significant new functions integrated into the Licensed Internal Code and Operating System/400. This includes new hardware support for the File Server I/O Processor (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.

With these enhancements the V3R1 system system software needs additional main storage to achieve the same performance level as a previous release. This is considered as the *working set size* of the system.

Approximately 650K additional bytes are required to maintain the machine pool (*MACHINE) page faulting rates you experience with V2R2, V2R3 or V3R0.5 systems. In most AS/400 system environments, this increase in size will have little or no impact on overall system performance.

Once you have installed V3R1 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 under 3.6.6, "Memory Guidelines" on page 28.

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

1. Set QFPRADJ system value to dynamic, run time tuning (CHGSYSVAL QPFRADJ(3))
2. Allow the tuning to take place over a 2 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 performance is equivalent to or better than the previous release.

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

However, any Version 2 or Version 3 Release 0.5 system that is approaching the following ″poor performance indicators″ should carefully analyze their current critical workload environment **before installing V3R1** and perform capacity planning analysis if significant additional work is considered after V3R1 installation:

- Page fault rates that approach the ″poor performance range″ as documented in Appendix A, "Guidelines for Interpreting Performance Data" on page 323 under tables Table 25, Table 26 on page 324, and Table 27 on page 324.
- Interactive application environment with CPU utilization above 70%
- Batch application environment with over 90% CPU utilization **and** batch job run time already close to the limit of a ″run time window″)

These are general recommendations for any new release installation, but are especially important for V3R1 installation on a system with small main storage sizes, such as 8MB or systems that should already be considering additional hardware resources.

If you anticipate additional system workloads after V3R1 installation, it is recommended you use the Performance Tools/400 BEST/1 support for capacity planning **before installing V3R1**.

If the current release is ″on the edge″ of poor performance, it is recommended you install at least 8MB of additional main storage. If you have considerable workload growth plans BEST/1 may recommend the addition of more resources.

See the next topic 3.6.6, "Memory Guidelines" on page 28 for more information on page fault guidelines.

3.6.6 Memory Guidelines

During the last quarter of 1994 new page faults per second guidelines were established for all V3R1M0 or earlier systems. Based on analysis of Performance Management/400 service offering performance data from over 1000 customers, the rates for ″good″ and ″acceptable″ performance are higher than in user documentation prior to V3R1.

Good and acceptable guidelines have been consistently higher for the medium speed and faster systems. However, the good/acceptable rates for these systems are now significantly higher for these models.

These new guidelines apply to all releases, beginning with V2R3 and are listed in the *Work Management for Version 3* **manual and Appendix A, "Guidelines for Interpreting Performance Data" on page 323** .

PTFs are available to V2R3, and V3R0M5 to enable the new guideline values to be used by the QPFRADJ automatic performance adjuster job, Expert Cache support, and the Performance Tools/400 functions such as the Advisor and BEST/1. These PTFs are:

```
o OPERATING SYSTEM/400:

       . V2R3  - SF17949
       . V3R0M5- SF17954


o PERFORMANCE TOOLS/400:

       . V2R2  - SF18188 and co-req's
       . V2R3  - SF17706 and co-req's
       . V3R0M5- SF18189 and co-req's
```

The co-requisite PTFs listed for the Performance Tools/400 Page Fault Guidelines update also include enhancements to BEST/1 disk modeling support. This includes:

- 9937-4xx models and the associated new controller with a 4MB Write Cache

  The new 9337 controller attaches to the 6501 IOP and also supports the 9337-2xx models

- The 6605, 6606, and 6607 disks that attach to the Advanced Series internal DASD IOPs 6502 and 6530 with 2MB Write Cache

See 6.11, "BEST/1 Capacity Planning Tool" on page 83 for additional information.

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 2 to 5 faults per second in the machine pool is now considered as delivering acceptable performance.

Refer to the *AS/400 Work Management Guide*, *AS/400 Performance Tools Guide* or Table 25 on page 323 for more information on paging guidelines.

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, then 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 ten more workstations or a new application, high page fault rates should **always** result in use of BEST/1 capacity planning to model the increased workload impact.

3.7 Disk

In most commercial applications, disk activity will probably be 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 vs 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

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, and checksum 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.

Table 11 on page 167 shows some Restore test results done by the laboratory with different levels of DASD protection. The following information is oriented toward interactive and batch applications and comparing the newer internal disks and external 9337 disks.

3.7.1 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 LIC 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 will increase 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 will cause 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, depending on the location of the data being written on the attached disk.

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 which 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 could happen in a system environment where there is an excessive number of disk I/Os.

Fields JBWIO and JBIPF are 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.

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, PRTCPTRPT, PRTTNSRPT and STRPFRMON commands 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).

3.7.2 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 LIC 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 LIC buffer and the ODP (Open Data Path).

Often, a logical disk I/O operation can be satisfied by data already residing in memory. On 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 will process approximately ″number-of-records″ 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 will be 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 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 somewhat 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 will do 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 LIC. The actual number of operations issued to a specific disk could 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.

3.7.3 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.

Based on laboratory analysis of disk IOP utilizations from customer performance data, V3R1 Performance Tools/400 BEST/1 and the Advisor support use new DISK IOP ″guideline″ (70%) and ″threshold″ (80%) values. The previous release values were 45% and and 50%, respectively.

Note the V3R1 performance documentation, such as the *BEST/1 Capacity Tool Planning - Version 3*, SC41-3341-00, lists the previous values of 45% and 50%. The documentation will be updated in the future.

3.7.4 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 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 will be 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 system can deliver quite acceptable performance for a rather heavy workload up to the 50%-55% busy range.

To determine disk busy usage, you use the performance monitor to collect disk I/O operations.

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 will increase to 1.00 second if the utilization increases to 40%.

Table 73 on page 353 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 index entries for more information on these controllers.

3.7.5 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% - 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 very 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.

3.7.6 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.

3.7.7 Disk Service Time Equation

The following 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 above equations, a single disk operation performed on a disk busy 30% of the time, with 6 users will take 0.039 seconds on 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 above formula does not take into account the larger write and read disk caches available on disks attached to the 6502, 6530, and 6501 disk controllers. There is no known formula to accommodate cache hit percentages.

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

Table 73 on page 353 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.

3.7.8 6502 Internal I/O Processor

With the introduction of the AS/400 Advanced Series, the 6502 and 6530 I/O Processors (IOPs) for internal DASD were introduced. Both of these IOPs have three SCSI buses capable of 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 and 6502 IOPs offer significant performance improvements over the older internal disk support when the new 1.03GB, 1.967GB, and 4.194GB disks are attached. The 6530 provides non-RAID support and the 6502 supports both RAID-5 and non-RAID support for attached disks.

The February 1995 *Performance Capabilities Reference* manual contains a large set of test results and should be reviewed for more details. In this redbook topic we discuss key performance considerations when using the newer 6606 (1.967GB) and 6607 (4.194GB) disks on the 6502.

The 6502 contains a non-volatile 2MB write-cache and supports up to 16 disks, depending on the tower features 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 per actual physical disk I/O issued by the system software (LIC). Only the 1.031GB, 1.967GB, and the 4.194BG disks make use of the 2MB non-volatile write-cache.

The 320MB, 400MB, and 988MB disks can be attached to the 6502 but cannot make use of the 2MB write cache or RAID-5. These non-RAID disks can coexst on the same 6502 with RAID-5 capable disks.

In addition, each 1.031GB, 1.967GB, 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 4 disks when the disk configuration is from 4 to 7 disks. With 8 disks configured for RAID-5, the checksum stripes can be spread over all 8 disks.

Table 73 on page 353 contains performance summary information for disks attached to the 6502 and 6530 controllers.

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

3.7.9 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 and the 9337-4xx DASDs 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 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 LIC.

With the new 6501 controller performance between the 9337-2xx and 9337-4xx models is essentially the same, but the 4xx models are of newer and more reliable (rated 1,000,000 hours versus 800,000 hours mean time between failures) technology.

Table 73 on page 353 contains performance summary information for disks attached to the 6501.

3.7.10 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 will meet the guideline of not exceeding 40%. The Ops/Sec/GB value plus the amount of DASD space required will allow you to select the acceptable DASD models to add or upgrade to.

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 (eg internal 2800s, 9336, 9337-025, 9337-220, etc.) 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 then 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.

The following sections provide a sample Disk Utilization report (Figure 5 on page 38) 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.

3.7.10.1 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. Follow the steps outlined below 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 command STRPFRMON INTERVAL(10) HOUR(1). Use the defaults for other command parameters. For additional information on collecting performance data, refer to the *AS/400 Work Management Guide* or to the *AS/400 Performance Tools/400 Guide* for more information.

3. If the system has Performance Tools 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 product is not installed, save the performance data to tape and restore it to a system with the Performance Tools installed.

After collecting the performance data needed to characterize the current DASD subsystem performance, continue with the instructions that follow Figure 5 on page 38 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                                          8/31/94 17:34:18
                                                           Disk Utilization                                               Page 0006
                                                           Thursday 8/23/94
Member . . . : THU082394   Model/Serial  . : B70     /10-27785  Main storage . . :   80.0 M   Started  . . . . : 08/23/94 08:33:53
  Library . . : COOK       System name . . :          RCHASM02  Version/Release  :   3/ 1.0   Stopped  . . . . : 08/23/94 09:33:49
```

Unit	Type	Size (M)	IOP Util	IOP ID	ASP ID	CSS ID	--Percent-- Full	Util	Op Per Second	K Per I/O	----- Average Time Per I/O ----- Service	Wait	Response
0001	9332	300	50.0	0-02	01	00	92.9	22.1	6.73	1.6	.032	.013	.045
0002	9332	300	50.0	0-02	01	00	90.8	17.9	5.23	1.9	.034	.011	.045
0003	9332	300	50.0	0-02	01	00	90.5	24.2	7.11	1.7	.034	.012	.046
0004	9332	300	50.0	0-02	01	00	90.5	28.2	9.01	1.8	.031	.022	.053
0005	9332	300	50.0	0-02	01	00	91.0	18.4	5.20	2.5	.035	.010	.045
0006	9332	300	50.0	0-02	01	00	90.5	21.8	6.19	2.3	.035	.011	.046
0007	9332	300	50.0	0-02	01	00	90.5	17.0	4.79	2.7	.035	.010	.045
0008	9332	300	50.0	0-02	01	00	90.6	24.0	7.07	1.7	.033	.015	.048
0009	9332	300	50.0	0-02	01	00	90.5	16.4	4.90	2.6	.033	.009	.042
0010	9332	300	50.0	0-02	01	00	91.1	18.3	5.37	2.2	.034	.011	.045
0011	9332	300	47.0	2-01	01	00	90.5	17.6	5.39	1.6	.032	.009	.041
0012	9332	300	47.0	2-01	01	00	90.5	29.9	8.38	1.2	.035	.016	.051
0013	9332	300	47.0	2-01	01	00	90.5	18.8	5.23	1.6	.035	.013	.048
0014	9332	300	47.0	2-01	01	00	90.6	18.0	5.80	2.9	.031	.008	.039
0015	9332	300	47.0	2-01	01	00	90.5	20.7	6.47	2.0	.031	.012	.043
0016	9332	300	47.0	2-01	01	00	90.5	14.7	4.42	2.5	.033	.012	.045
0017	9332	300	47.0	2-01	01	00	90.6	26.8	8.15	3.2	.032	.021	.053
0018	9332	300	47.0	2-01	01	00	90.6	15.4	4.45	1.0	.034	.009	.043
Average							90.7	20.5	6.11	1.2	.032	.012	.044

```
**************************************************************************************
*  TOTAL          5400                                                       109.89  *
*                    1                                                           2   *
* NOTE: Totals are not on the report. They must be calculated.                       *
**************************************************************************************
```

Unit	--	Disk arm identifier
Type	--	Type of disk
Size (M)	--	Disk space capacity in millions of bytes
IOP Util	--	Percentage of utilization for each Input/Output Processor
IOP ID	--	Input/Output Processor identification number (Bus-IOP)
ASP ID	--	Auxiliary Storage Pool ID
CSS ID	--	Checksum Set 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 5. Performance Tools System Report, Disk Utilization

1. Refer to the ″Disk Utilization″ section of the System Report. From this report the following data can be obtained:
   - 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)″ colum 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 2 when calculating the sum. In the example ″Disk Utilization″ report, the total number of GBs would be 5.400 GB. **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 2.

   In the example ″Disk Utilization″ report the total operations per second would be 109.89 **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 20.4 Ops/Sec/GB.

3.7.10.2 Select the DASD model that meets your needs

You can then use the Ops/Sec/GB value to determine what model of DASD best fits your current or projected DASD subsystem performance requirement. If you require internal DASD, refer to the 3.7.11, “AS/400 Internal DASD Subsystem Ops/Sec/GB Chart” on page 40. If you require external DASD, refer to the 3.7.12, “AS/400 9406 9337 DASD Subsystem Ops/Sec/GB Chart” on page 40.

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 18-20 range.

To help you read the bar charts in the following figures, examples of the highest Ops/Sec/GB values for specific disk models are listed below:

- 6502 IOP: non-RAID 6602 disk = 34
- 6502 IOP: RAID 6602 disk = 28
- 6501 IOP: non-RAID 9337-440 = 22
- 6501 IOP: RAID 9337-440 = 20

In this example, assuming you required internal DASD and required RAID protection, then you would go to 3.7.11, “AS/400 Internal DASD Subsystem Ops/Sec/GB Chart” on page 40 and compare the ops/sec/GB ranges for the DASD models attached to the 6502 RAID mode. In this case the model that meets the requirement of 20.4 ops/sec/GB is the 6602.

3.7.11 AS/400 Internal DASD Subsystem Ops/Sec/GB Chart

This section shows acceptable performance range of disk I/O operations for internal DASD offered on the AS/400 models. Use the highest disk I/O operations for each bar when considering peak activity.

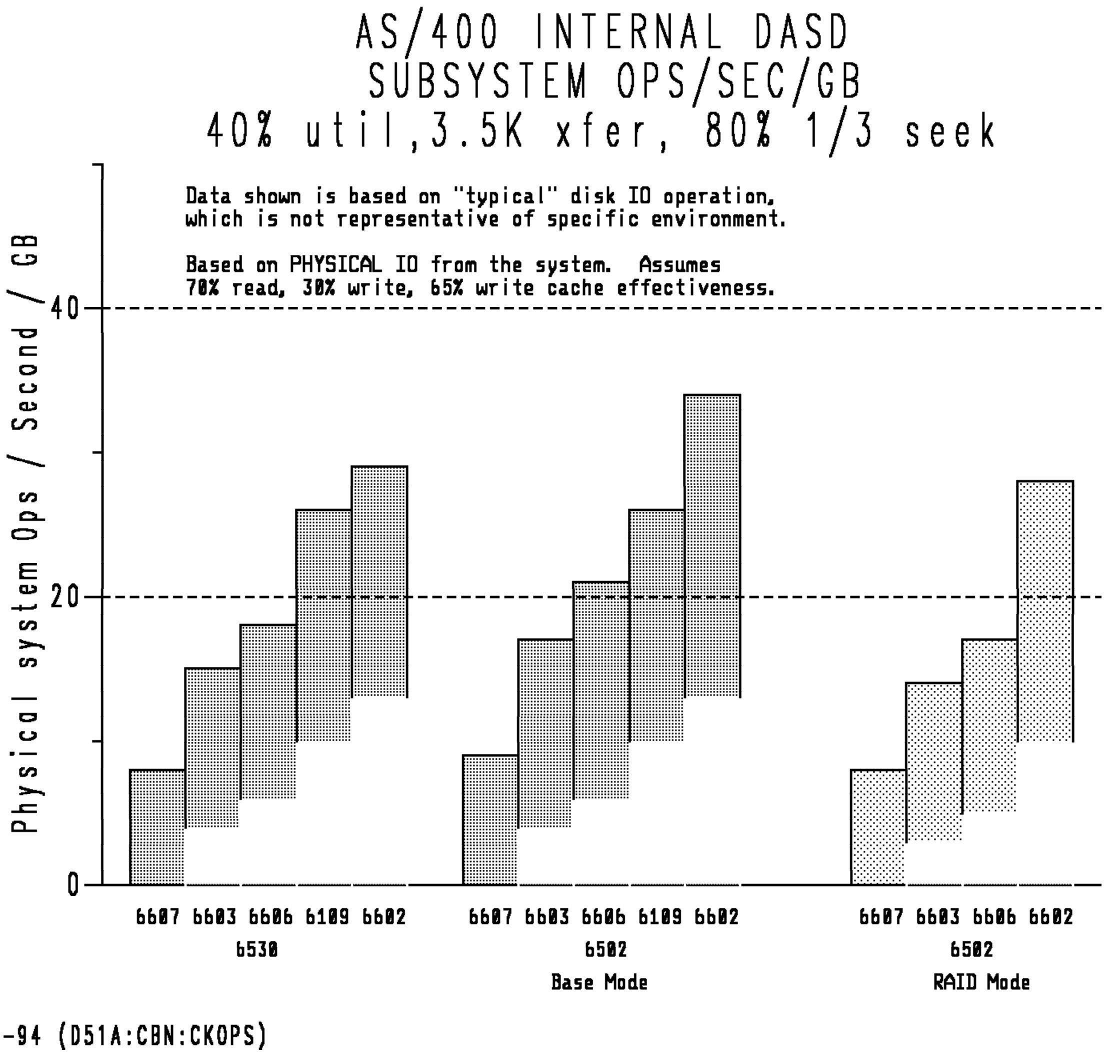
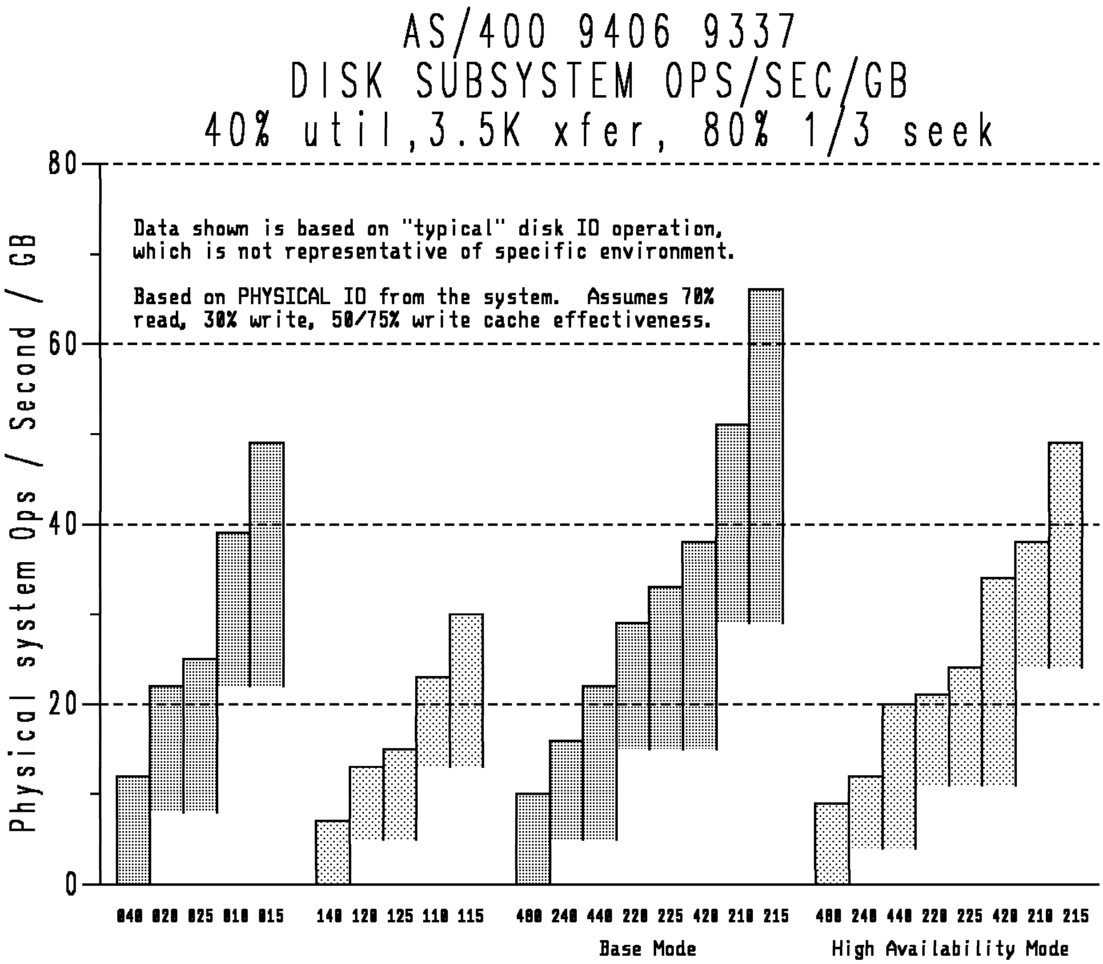


Figure 6. Internal DASD Operations/Second/Gigabyte

3.7.12 AS/400 9406 9337 DASD Subsystem Ops/Sec/GB Chart

This section shows acceptable performance range of disk I/O operations for external DASD offered on the AS/400 models. Use the highest disk I/O operations for each bar when considering peak activity.



09-09-94 (D51A:CBN:9406OPS)

Figure 7. 9337 DASD Operations/Second/Gigabyte

The 6502, 6530 internal IOPs and the 6501 external disk IOPs have sufficient processor capacity to support up to 40% disk arm utilization for the maximum supported disk configuration. If you are configuring the older 9337-0xx or 9337-1xx devices you need to also consider that 6500 IOP can support a maximum of 120 I/Os per second for *all attached disks*.

Since there may be more than one set of disk models that satisfy your current disk configuration Ops/Sec/GB, you need to factor original cost of upgrade, disk reliability, and any future growth in DASD storage requirements.

The Performance Tools/400 BEST/1 capacity planning support is especially useful in your disk capacity planning efforts.

3.8 Hardware Bus Data Transfer Rate Considerations

The AS/400 hardware bus maximum megabyte data transfer rates were a concern on the Version 1 hardware models if "too much" disk, tape, or communications work was done on a single bus.

This concern was removed with the Version 2 (models D-F) and Advanced Series models except for 3490 tape device attachment as discussed here.

The Version 1 hardware supported 5M bytes per second for Bus 0, 2.5M bytes for Bus 1 and 1.75M bytes for Bus 2. Version 2 D-E model systems support bus speeds of 8M bytes per second for all buses.

The F model series and Advanced System Series 2xx models and 3xx models support:

- Copper bus maximum sustained speed of 8M bytes per second (bus 0)
- Fiber optic bus maximum sustained speed of 27.75M bytes per second (buses 2-7)

Using the F and Advanced model series systems as an example, there is no maximum speed concern except where 3490 tape drive and system disks are placed on the same bus. Where the customer requires the 3490 tape devices to run at maximum speeds, they must be on separate buses from those with disks attached. With both the 3490 tape and disk on the same bus, there is a chance for performance problems since typically the disks and tape drives are being heavily accessed at the same time.

3.9 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:

- The I/O Processor (IOP)
- The line
- The control unit
- The 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 operation of the printer.

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

3.9.1 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 LIC 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 will cause 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.

3.9.2 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 323 for more information.

3.10 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.

3.10.1 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 323 for more information.
- Changing the activity level value will 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 would overload the system and cause 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 *AS/400 Performance Tools Guide* or Appendix A, "Guidelines for Interpreting Performance Data" on page 323 for information on setting activity level values.

3.10.2 Job Priority Adjustments

The AS/400 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. Priority is returned to its previous value at the end of the conflict. You may observe this priority adjustment with the WRKSYSACT (Work with System Activity) command.

Also, if a job has used between 5% to 10% of CPU in its last eight I/O requests, its run priority is increased to allow it to run more quickly after the I/O operation. If the amount of CPU utilized was less than 5%, the priority is raised even higher. Once these jobs with adjusted priority begin to use more than 10% of the CPU, their priority is adjusted downwards again. The intent of these adjustments is to keep disk bound jobs from having to wait too long for CPU resources. This allows the system to maximize the amount of data flowing to and from disk, memory and the CPU, improving total system throughput. This disk priority adjustment is not visible on any OS/400 displays or performance reports.

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

3.10.3 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, then 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 2 seconds for interactive jobs and 5 seconds for batch jobs may often be too high, especially for the high-end processors. As an initial value, set the time slice at 3 times the average CPU seconds per transaction.

3.10.4 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, the condition is called a lock conflict if it occurs based on an operating system function. If it occurs on Licensed Internal Code (LIC) resources, 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, 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 could 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 very short periods of time, such as .1 second or less. It is when these lock or seize waits become greater than .2 seconds or there is a long series of small waits that performance impact is degraded enough to do 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 would occur at CPU utilizations less than 90%.

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

UNIX-Processing Resource Conflicts

Note that with V3R1 and the AS/400 CPA Toolkit technology preview support, UNIX applications may be ported to the AS/400 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.

3.10.5 Activation Groups and the Integrated Language Environment

The Integrated Language Environment (ILE) was introduced to the AS/400 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 and COBOL, and the other high level languages in the initial releases of AS/400. 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.

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 very close to calls to internal routines within a single large program.

Note performance of a single large monolithic program in V3R1 versus V2R3 should be approximately the same if no coding changes are made.

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

With V3R1, Control Language and RPGIV/400 support the ILE capabilities. ILE for COBOL/400 should be available sometime in 1995.

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.

AS/400 provides several manuals that include ILE information. See the AS/400 *Publications Ordering*, SC41-3000-01, to determine what formal ILE documentation is available. The following redbooks also provide very 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

Chapter 10, “Design and Coding Tips” on page 189 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.**

3.10.5.1 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 name will appear in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Start Sample Address Monitor (STRSAM) output (if requested), and the Timing and Paging Statistics Tool (TPST) PRPQ output (if requested).

3.10.5.2 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 module name will appear in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Start Sample Address Monitor (STRSAM) output (if requested), and the Timing and Paging Statistics Tool (TPST) PRPQ output (if requested).

3.10.5.3 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 recreation. The modules they contain can be changed without rebinding the bound program under certain conditions. See HLL specific documentation for details.

One or more service programs are ″bound″ to a mainline program via 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.

The program name will appear in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Start Sample Address Monitor (STRSAM) output and the Timing and Paging Statistics Tool (TPST) PRPQ output

3.10.5.4 ILE Bound Programs

A bound program is a set of one or more *modules* and/or *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)**.

The program name will appear in trace job output, the Performance Tools/400 Transaction Report output, the Performance Tools/400 Start Sample Address

Monitor (STRSAM) output and the Timing and Paging Statistics Tool (TPST) PRPQ output

3.10.5.5 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/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
- unknown until ILE COBOL/400 availability

3.10.5.6 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/modules in the stack is ″percolated″ up to the control boundary stack entry.

A procedure will be 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.

3.10.5.7 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 could be 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 support activation groups with their Version 3 Release 1 availability.

Programs within a job and files opened by those files are ″assigned to an activation group″ either explicitly or implicitly. Programs created in releases prior to V3R1 default to the job *default activation group.* Programs created in V3R1 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), Create Bound CL Program (CRTBNDCL), and Create Bound RPG Program (CRTBNDRPG) commands 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 V3R1. Figure 8 on page 50 shows an example of DSPJOB command ″display call stack and display activation group″ output for programs created prior to V3R1. Figure 9 on page 50 and Figure 10 on page 51 show similar output for ILE programs created on V3R1.

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: RCHASM02
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: RCHASM02
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 on page 51 shows this information for the RPG example.

```
                              Display Call Stack
                                                          System: RCHASM02
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 shows ILE program CSR1 was dynamically called.

2 For ILE RPG/400 and ILE CL, this procedure name will always be 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.

3.11 Non-Interactive or Batch Jobs

Non-interacitve jobs are sometimes overlooked in a performance and throughput review. As previously discussed under 3.4.4, "Server Model Series Considerations" on page 19, 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 file blocking isn′t changed, the logical disk I/Os will be the same for the different runs. However, the physical I/Os can vary between runs due to other jobs in the system which may cause page stealing etc.

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 has commenced execution, are:

- CPU time
- Disk I/O time

A simple formula for batch job run time is shown below. An example of estimating job time from this formula is contained in 6.12, "Manual Batch Run Time Estimation" on page 87.

```
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. Im many cases the values in JBIPF and JBWIO are rather small, but only a query can verify this.

The Performance Tools System Report provides average values for this information. A query on Performance Monitor file QAPMJOBS will 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.

3.12 Other References

The *Performance Tools Guide* and *Work Management Guide* provide additional information on performance tuning. For additional performance information refer to the “Related Publications” in this publication.

Chapter 4. Performance Management Methodology

4.1 Performance Management Overview

This chapter presents the methodology for performance management. A flowchart is used to describe the organization of information in Chapters 5 through 7 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 the reader already has a formal performance management methodology, they may consider integrating portions of the methodology presented herein with their existing procedures.

A compressed image of the following flowchart appears at the start of Chapters 5 through 7. This is to make it easier to reference information presented in each of those chapters to procedures discussed here for the performance management flowchart.

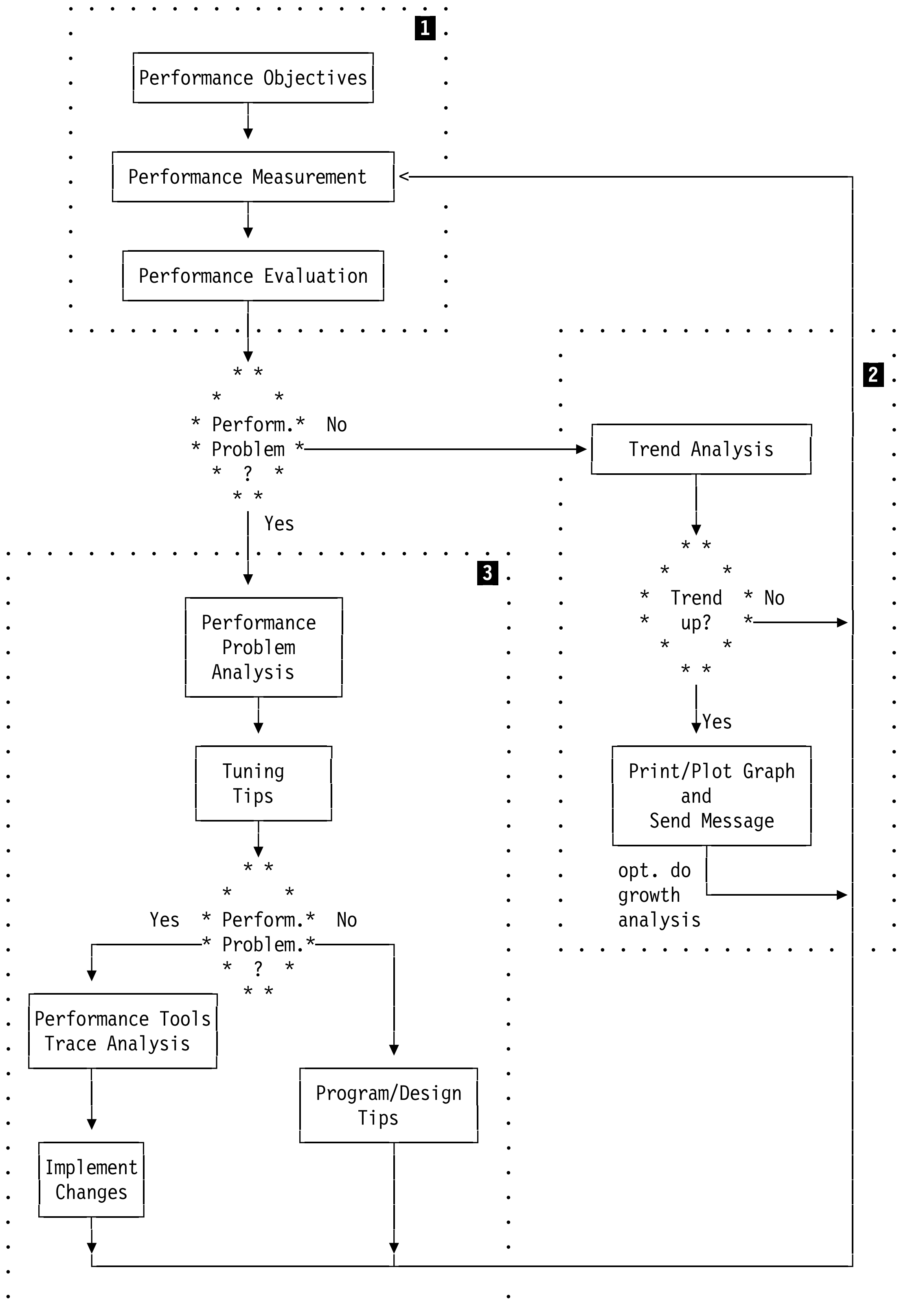


Figure 11. Performance Management Flowchart

4.2 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 5, “Performance Management and Review” 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, definition of an interactive transaction and the expected transactions per hour. Definition of a batch job throughput could be based on a time period and records processed or lines of printed output during that time period. In a communications environment, the objective could be 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 his current configuration. If a purchased application package will be 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 he expects for each type of job (for example, response time for local and remote terminals, throughput for batch jobs, office tasks per hour, etc.).

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, etc. The customer should define his 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 screen 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 which 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 very 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. Chapter 7, "Performance Problem Analysis" 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, then 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, then 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."

4.3 Performance Trend Analysis

Stage **2** of the performance management flowchart is discussed in the following paragraphs. Chapter 6, "Performance Trend Analysis" provides additional details.

Trend analysis involves the review of performance measurement data collected over a period of time and looking 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 command (DSPHSTGPH). Historical Data can be created using 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 would *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 then projecting what hardware upgrades will be 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.

4.4 Performance Problem Analysis

Stage **3** of the performance management flowchart is discussed in the following paragraphs. Chapter 7, "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 and the Work with System Status (WRKSYSSTS) command and the Performance Tools Advisor function (ANZPFRDTA command or menu options).

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

Review Performance Tools Advisor output first as it is a ″fast path″ through performance monitor *sample data* for the Advisor using the Performance Tools/400 (5763-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 1 Performance Tools/400 (5763-PT1) licensed program. The Advisor function outputs 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 very 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 command (SBMJOB) 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 or WRKACTJOB commands and can be used interactively to identify system and user jobs consuming significant resources, which may assist in identifying functions causing poor performance.

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, etc. These values should be compared with those obtained from the Performance Tool reports. Any discrepancy will indicate 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 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 use of 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, PRTJOBTRC. Job trace output may identify which user programs or IBM programs/modules are called and which of these programs are consuming large amounts of CPU or causing a large number of page faults.
- Use of the Timing and Paging Statistics PRPQ program (5799-EER) functions to list detailed CPU usage and disk accesses by programs, Licensed Internal Code (LIC) tasks, and modules.
- Sample Address Monitor (SAM) support (STR/ENDSAMCOL and STR/ENDSAM commands) for identifying high CPU processing time instructions within a program.
- Print object lock support (PRTLCKRPT command) for further refinement of object seize/lock holders and waiters.
- Analyze data base file and program relationships and usage (ANZACCGRP, ANZDBF, ANZDBFKEY, ANZPGM commands).
- Analyze object and disk storage usage (STR/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 may lose some of its storage. SETOBJACC is used to cause no page faulting to occur in the pool, and QPFRADJ would consider 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 will recommend storage pool reassignment and make the change if requested by operator response. The Advisor analyzes performance data which was 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 systems, but you cannot have the Advisor change your system since the recommendations are not for the same machine. Another reason why the Advisor would not make any changes would be if the pool configuration has changed since the performance data was collected, or if the automatic performance adjustment facility was was turned on using QPFRADJ values.

The Advisor will 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 9, “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, then additional analysis must be performed, by collecting the performance data with the option TRACE(*YES). This option allows 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.

The Advisor under Version 2 Release 2 Performance Tool supports 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/off, response time, CPU seconds, disk I/O operations, object lock/seize times, time slice end, short wait times, etc. If the Transaction report shows seize/lock conditions, the Print Lock Report should be reviewed.

Consider the use of 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.

The Timing and Paging Statistics (TPST) program, 5799-ERR, may be used to identify which programs to investigate. TPST output shows which programs, processes, tasks, and modules are using large amounts of CPU time or numbers of disk accesses. Chapter 8, “Additional Performance Tools” provides more information on TPST.

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, etc. 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 Appendix B 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, then the Performance Tools Sampled Address Monitor (SAM) support should be used to identify important program instructions. SAM output shows the relative amount of processing time spent in different parts of a program or a set of programs.

Chapter 10, “Design and Coding Tips” of this publication has a number of application design tips for programming that will assist in maximizing performance.

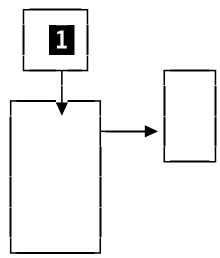
In some cases the tips listed can be used to improve the performance of existing applications by merely making a few small changes. However, in many cases integrating the tips into existing applications can be time consuming and exposes the application to regression problems. In those cases adding hardware 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 5. Performance Management and Review

5.1 Content

This chapter discusses stage **1** of the performance management flowchart.



5.1.1 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.

  Chapters 6 and 7 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 Chapters 6 and 7.
- A description of the initial analysis made with this data.

5.2 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 will be acceptable to your local and remote workstations.

You should think about how long your regular batch processes will 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 screens in BEST/1.

If the application is already available, Performance Monitor output can be used to initially help establish a base set of statistics, which should then 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 will almost always be 1-2 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 will be run and their expected duration
- A list of other unscheduled batch jobs that may be required

5.3 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 will give 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.

5.3.1 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 is one suggestion of using 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, etc. 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 then 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 will indicate the trace time period under the System Summary Data heading.

5.3.2 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 menu (WRKPFRCOL) 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.

5.4 Performance Review Options

5.4.1 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 1 and pressing the Enter key. The Advisor will analyze performance data at that particular interval and give 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 5 in the Option column.

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

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

You can refer to 7.6.11, "Advisor" for further information on Advisor.

5.4.2 Performance Graphics

Performance data collected by Performance Monitor can be displayed in graphical format using the Performance Graphics option in Performance Tools/400. The graphs can be displayed interactively or 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 froma 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.

5.4.3 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, then summarizes and transmits the summarized data to the AS/400 Competency Center in Rochester weekly.

PM/00 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. Chapter 8, "Additional Performance Tools" provides more information on PM400.

PM/400 does not require AS/400 Performance Tools (5738-PT1) and has no intention to replace that product.

5.4.4 Performance Tools Report

Printing performance reports provides complete information on collected performance data. You can review the performance of specific jobs or transactions, or other performance elements. This can be done by taking Option 3 (Print Performance Report) from the AS/400 Performance Tools menu which will lead you to 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 via 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 require the first 2 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
  _      Q922160900   Data collected 080392 at 0900       08/03/92   09:00:38
  _      Q922130900   Data collected 073192 at 0900       07/31/92   09:00:42
  _      Q922120900   Data collected 073092 at 0900       07/30/92   09:00:43
  _      Q922101447   Data collected 072892 at 0900       07/28/92   14:47:34
  _      Q922090900   Data collected 072792 at 0900       07/27/92   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 RCHASM02 KB
```

Figure 12. 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 RCHASM02 KB
```

Figure 13. Select Catagories 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 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 will find the basic set of information to compare against your *performance objectives* and/or guidelines tables as shown in Table 24 on page 323.

- 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 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 24 on page 323 for other CPU types.
- Check the percent of space in use and the utilization on Disk Utilization part of report; compare those values to Table 24 on page 323 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.
- Avg Util and Max Util column on Communication part gives you the average and maximum percentage of line capacity used during the measured interval. Compare those values to Table 24 on page 323.

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, or if the same disk unit appears in the high utilization, or to identify the pool which 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 24 on page 323. Wait-to-ineligible need not be 0 all the time, but must be less than .25 for good performance.
- Disk Activity shows average disk activity per hour, and 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 dedicated batch, performance is not normally degraded. However, if there are lots of interactive jobs, high disk utilization can indicate a performance problem.
- Data Base Journal Summary includes 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 7.5.1, "Interactive Analysis" of this publication 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 7, "Performance Problem Analysis" on page 91, provides details for analyzing performance data.

If you have no indication of any problems with system performance, proceed to Chapter 6, "Performance Trend Analysis" in this publication.

5.5 Performance Data Conversion

Each new release of the AS/400 may require conversion of performance data collected on a previous release. Performance data collected with STRPFRMON or WRKPFRCOL from a previous release machine needs to be converted before analyzing. The CVTPFRDTA command converts your performance data library from 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.

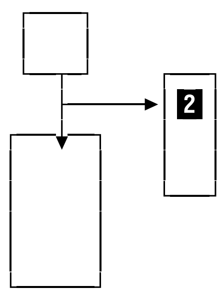
After all performance data have been successfully converted, other Performance Tool functions can be performed on the new release.

Chapter 6. Performance Trend Analysis

6.1.1 Content

This chapter expands on the topic of trend analysis introduced as Stage **2** of the performance management flowchart in Chapter 4, “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.



6.1.1.1 Major Topics Covered in This Chapter

- Why Do Trend Analysis?
- When Should Trend Analysis Be Done?
- Trend Analysis Methodology
- Query Measured Profile Data
- Performance Graphs
- Trend (Historical) Graphs
- PM400
- Performance Investigator
- Archiving Performance Data
- BEST/1 Capacity Planning Including batch run time analysis
- Comparing Extrapolated Data with Historical Data
- Notification of Performance Trends

6.2 Why Do Trend Analysis?

Even if you currently have no performance concerns, it would be beneficial to be warned 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 will not be 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.

6.3 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, then a more frequent review may be indicated. If a problem is encountered, identifying and resolving the problem will take precedence over trend analysis activity, but will not replace it. Trend analysis is an insurance against the sudden discovery of a capacity blow-out.

6.4 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 5, "Performance Management and Review," a subset of the data may be retained for trend analysis. Alternatively, you could 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 will be no need to save this much information.

Utilization trends can be detected in most of the system resources, like 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.

6.5 Query Measured Profile Data

Using the System Report (option 1) of the Print Performance Report Menu or the Print System Report (PRTSYSRPT) command, it is possible to create a measured profile of the average workload for the selected period. This writes a single record to the file QACPPROF. This file contains fields representative of the transaction workload and the utilization of some of the key system resources.

This technique is also used in the main line program for Automatic Performance Management Tool (APMT) discussed in Appendix C, "IBM Internal Use Only Tools/Documents" on page 409, where a profile is created for every measurement collected.

In this section we have simulated the effect of observing performance measurements, by creating measured profiles on data gathered over many periods of time.

The following pages discuss an example query of the records in file QACPPROF, which illustrate the kinds of performance trend analysis that can be done with this file.

We have also used the graphics facility of the Performance Tools to create a compound graph. The graph shows the transaction response time, superimposed on bar charts which show the percentage CPU use for system, interactive and batch work. Each graph is over all the records shown in the query listing.

The purpose in providing these tables and graphs is to show how this information **can detect trends** that may be occurring in the performance of your computer system.

07/02/91 18:04:15 File....QACPPROF Library....FE21 PAGE 1

Profile Name	Tot AWS	Trans Per Hour	Resp Time Secs	Excp Wait Var.	CPU Sec/ Tran	CPU Util I+S	CPU Util Btch	Tot CPU	Batch Impact Factor	Nbr Bch Job	Disk Arm Util	DIO/ Tran Int.	DIO/ Tran Sys.	Batch DIO/ Sec	Work Set Size	OCR
P01-0645	8	2,191	.8	.78	.1540	.14	.18	.32	.627	8	.03	7.0	6.2	42.3	616	.020
P02-0650	7	2,131	.7	.59	.1364	.09	.01	.10	1.079	2	.01	7.7	3.3	2.7	532	.020
P03-0655	5	1,661	1.4	1.32	.1579	.09	.01	.10	1.509	2	.01	10.1	11.7	2.3	610	.030
P04-0700	4	830	.9	.49	.1893	.08	.32	.40	.213	7	.09	13.8	74.6	87.8	623	.030
P05-0705	4	927	.6	.14	.1538	.08	.35	.43	.143	4	.08	14.5	23.4	95.6	527	.020
P06-0710	5	785	.4	.16	.1403	.06	.24	.30	.219	4	.06	10.6	36.4	64.9	499	.030
P07-0715	5	505	.8	.20	.2383	.05	.02	.07	4.960	4	.01	25.1	12.6	2.1	702	.030
						1						2				
P08-0720	8	951	1.8	.00	1.0911	.34	.01	.35	4.951	2	.05	189.8	7.3	.8	790	.040
P09-0725	7	794	5.0	.08	.8355	.26	.01	.27	2.459	3	.07	268.3	54.0	1.8	603	.020
P10-0730	9	1,565	1.0	.31	.1336	.08	.04	.12	.777	3	.02	23.2	11.3	5.7	410	.020
P11-0735	16	1,740	2.6	.19	.5997	.32	.03	.35	3.292	5	.05	114.8	22.2	3.7	871	.080
P12-0740	14	2,420	2.3	.28	.5369	.41	.03	.44	2.453	5	.05	79.7	28.9	3.7	781	.060
P13-0745	17	2,492	1.2	.29	.3346	.27	.08	.35	.601	4	.05	57.7	13.7	8.7	644	.070
P14-0750	19	3,335	2.0	.37	.3184	.36	.17	.53	.176	5	.09	61.6	15.0	44.1	528	.060
P15-0755	19	2,311	2.4	.32	.4341	.35	.27	.62	.506	6	.07	68.5	19.4	43.0	611	.060
P16-0800	20	3,154	.8	.32	.1935	.23	.16	.39	.765	7	.04	18.6	15.2	20.4	606	.070
P17-0805	20	3,130	.8	.27	.1876	.21	.10	.31	1.734	7	.03	21.4	8.7	8.3	571	.070
P18-0810	25	5,273	.5	.06	.1791	.30	.07	.37	2.596	8	.03	10.1	4.2	5.9	606	.090
P19-0815	29	4,346	1.0	.19	.2380	.36	.17	.53	.664	6	.07	24.0	9.5	36.5	711	.130
P20-0820	32	4,238	.9	.28	.2335	.35	.20	.55	.171	5	.08	19.8	12.9	49.5	708	.170
P21-0825	38	3,575	1.2	.78	.1988	.27	.33	.60	.376	5	.06	16.1	16.1	28.8	620	.190
P22-0830	46	4,490	1.0	.43	.1955	.29	.18	.47	.283	5	.06	16.7	7.3	33.8	644	.230
P23-0835	53	8,584	1.0	.44	.1679	.45	.12	.57	.545	6	.06	15.0	4.9	13.9	566	.230
P24-0840	67	9,102	1.8	1.17	.1816	.53	.33	.86	.164	7	.08	13.4	6.7	38.2	600	.240
P25-0845	76	10,667	1.3	.80	.1575	.59	.07	.66	.559	3	.06	10.0	10.4	7.4	559	.260
P26-0850 **	70	11,064	1.6	.96	.1675	.64	.26	.90	.172	7	.07	9.6	10.9	20.0	579	.310
P27-0855	85	9,680	2.2	1.37	.2064	.64	.28	.92	.130	9	.11	16.4	9.7	43.2	661	.420
P28-0900	72	9,138	2.1	1.39	.1975	.57	.31	.88	.142	8	.09	17.1	8.7	32.1	638	.340
P29-0905	69	10,101	2.3	1.39	.1997	.64	.24	.88	.430	10	.09	18.6	8.5	25.0	641	.340
P30-0910	81	11,304	2.5	1.71	.1873	.67	.26	.93	.308	13	.10	14.9	8.8	39.9	625	.370
P31-0915	67	9,246	2.6	1.96	.1964	.58	.36	.94	.000	10	.10	12.8	9.7	57.1	649	.310
P32-0920	68	8,945	2.4	1.70	.1919	.55	.36	.91	.228	8	.10	18.0	8.1	53.6	644	.320
P33-0925	75	11,305	2.0	.85	.1617	.57	.23	.80	.320	6	.06	10.3	5.3	19.5	558	.320

Figure 14. Query on Measured Profiles P01 to P33

1 Most of the records in this block show a very high CPU time per transaction for an AS/400 B70 system, and a CPU time per second that is much higher than the other records in the listing. This is probably caused by running batch processes interactively, or by the presence of very complex interactive transactions which results in the relatively poor response time for such a low number of transactions.

2 Note the much higher number of disk I/O operations for interactive transactions for these same time intervals, which further supports the identification of batch type processing run at interactive priority or complex (many disk I/O) interactive transactions. Note that there are relatively few active work stations during this time.

07/02/91 18:04:15 File....QACPPROF Library....FE21 PAGE 1

Profile Name	Tot AWS	Trans Per Hour	Resp Time Secs	Excp Wait Var.	CPU Sec/ Tran	CPU Util I+S	CPU Util Btch	Tot CPU	Batch Impact Factor	Nbr Bch Job	Disk Arm Util	DIO/ Tran Int.	DIO/ Tran Sys.	Batch DIO/ Sec	Work Set Size 5	OCR
P34-0934	74	11,617	2.4	1.65	.1854	.67	.25	.92	.963	25	.09	12.9	6.4	25.8	624	.350
							3									
P35-0939	78	11,835	2.8	1.77	.1927	.71	.24	.95	.000	7	.09	15.9	6.4	26.4	642	.380
P36-0944	78	12,268	3.0	1.34	.2195	.83	.12	.95	.000	9	.09	15.6	8.8	11.7	720	.430
P37-0949	85	11,895	4.8	3.46	.2269	.81	.12	.93	.000	11	.07	12.1	5.4	12.8	735	.390
P38-0954	88	11,690	5.1	3.45	.2368	.83	.14	.97	.000	9	.07	12.5	5.5	14.4	754	.500
P39-0959	83	11,582	4.3	2.92	.2287	.79	.17	.96	.000	10	.08	14.4	5.6	18.5	718	.460
P40-1004	82	9,258	5.5	4.01	.2770	.78	.15	.93	.023	9	.08	13.1	6.1	11.2	818	.530
P41-1009	67	7,681	5.7	4.38	.2993	.71	.20	.91	.091	10	.08	16.3	10.5	21.7	872	.460
P42-1014	89	9,608	3.9	2.80	.2362	.71	.23	.94	.000	14	.08	13.2	10.4	27.1	738	.510
P43-1019	92	9,764	4.0	2.98	.2262	.70	.25	.95	.000	14	.10	16.2	11.5	33.2	715	.500
P44-1024	97	10,500	2.7	1.92	.1776	.61	.28	.89	.346	16	.10	12.0	11.4	39.2	586	.420
P45-1029 **	85	10,524	2.3	1.44	.2175	.71	.21	.92	.175	11	.07	12.4	7.2	19.7	707	.430
P46-1034	81	8,837	2.2	1.59	.1839	.55	.31	.86	.320	10	.09	13.8	12.7	26.5	608	.360
P47-1039	84	8,885	2.7	1.82	.2219	.63	.31	.94	.000	9	.08	13.5	11.3	31.5	714	.350
P48-1044	87	8,524	2.0	1.29	.2060	.56	.31	.87	.083	5	.09	19.9	10.0	35.2	659	.440
P49-1049	79	8,969	1.2	.74	.1662	.47	.10	.57	.443	4	.05	11.6	7.6	10.9	590	.370
P50-1054	91	11,221	1.9	1.27	.1884	.64	.18	.82	.459	8	.06	10.8	5.6	19.5	630	.450
P51-1058	95	11,353	1.6	1.11	.1601	.57	.23	.80	.373	7	.07	9.6	7.3	21.4	576	.440
P52-1108	124	11,677	3.3	2.55	.1896	.69	.26	.95	.000	12	.08	12.4	7.3	28.9	632	.620
							4									
P53-1113	101	10,607	6.8	6.00	.2181	.66	.28	.94	.000	13	.08	12.8	6.2	30.0	706	.570
P54-1118	109	9,018	14.8	14.26	.1722	.75	.19	.94	1.470	15	.05	11.3	4.9	16.6	584	.400
P55-1123	123	8,717	15.1	14.79	.1874	.82	.10	.92	2.880	15	.04	11.0	5.0	11.1	631	.490
P56-1128	119	9,379	15.4	15.04	.1892	.83	.11	.94	1.711	13	.05	12.4	5.1	11.6	631	.590
P57-1133	115	7,031	19.0	18.84	.1856	.87	.08	.95	2.429	12	.07	12.4	9.4	8.6	628	.560
P58-1138	139	7,573	24.1	23.51	.1784	.81	.13	.94	1.880	13	.07	13.2	9.3	12.4	599	.650
P59-1143	138	11,931	18.6	18.18	.1754	.82	.09	.91	2.409	16	.10	12.9	9.0	11.8	600	.510
P60-1148	128	13,268	7.7	6.78	.1847	.76	.14	.90	1.046	14	.10	12.5	6.5	14.1	612	.480
P61-1153	136	13,882	10.1	8.89	.1849	.79	.13	.92	1.218	17	.10	12.3	7.5	14.8	620	.510
P62-1158	133	12,834	9.8	8.81	.1994	.78	.11	.89	1.808	17	.09	17.6	5.4	12.5	651	.680
P63-1203	121	13,677	5.3	4.41	.1705	.70	.21	.91	.683	18	.08	13.1	4.9	20.8	574	.530
P64-1208	104	10,405	5.3	4.84	.1914	.64	.31	.95	.000	18	.08	13.4	5.6	25.8	640	.670
P65-1213	92	9,842	4.5	3.90	.1789	.58	.38	.96	.000	13	.09	12.3	11.0	30.5	610	.390
P66-1218	92	10,029	5.3	4.58	.1915	.61	.33	.94	.255	16	.09	14.7	8.7	26.2	647	.420
P67-1223	102	9,138	4.5	3.84	.2072	.60	.36	.96	.000	14	.10	14.6	8.6	43.3	689	.500
P68-1228	97	10,715	3.2	2.52	.1774	.61	.35	.96	.000	14	.10	14.2	8.8	44.1	607	.430

Figure 15. Query on Measured Profiles P34 to P38

3 All these measurements show that the CPU utilization is at the upper threshold limit of 70% or higher for system and interactive use, resulting in unpredictable and poor response times.

4 These measurements show poor response times where the variable exceptional wait time is extremely high, and the percentage CPU is also consistently above the upper threshold limit. It will need detailed analysis to determine the cause of poor response.

5 indicates ″working set size.″ As recorded in file QACPPROF, this value is an estimate that is less accurate in releases after V2R2. See index entry ″working set″ for the recommended process for determining working set size.

The exceptional wait time is generally very high for this system, which means that there could be excessively long wait times for record locks, seize logical files, message queue or data area locks. Other possibilities include waiting for remote communications operations to complete or some other application design consideration.

In conclusion, it would appear that the knee of the performance curve for this particular system occurs between 10,000 and 12,000 transactions per hour, at which throughput, the average response time appears to be in excess of three seconds, and unpredictable.

6.6 Performance Graphs

This is a function of the AS/400 Performance Tools program product which is used for graphically displaying performance data. It requires performance data to be collected using the STRPFRMON command.

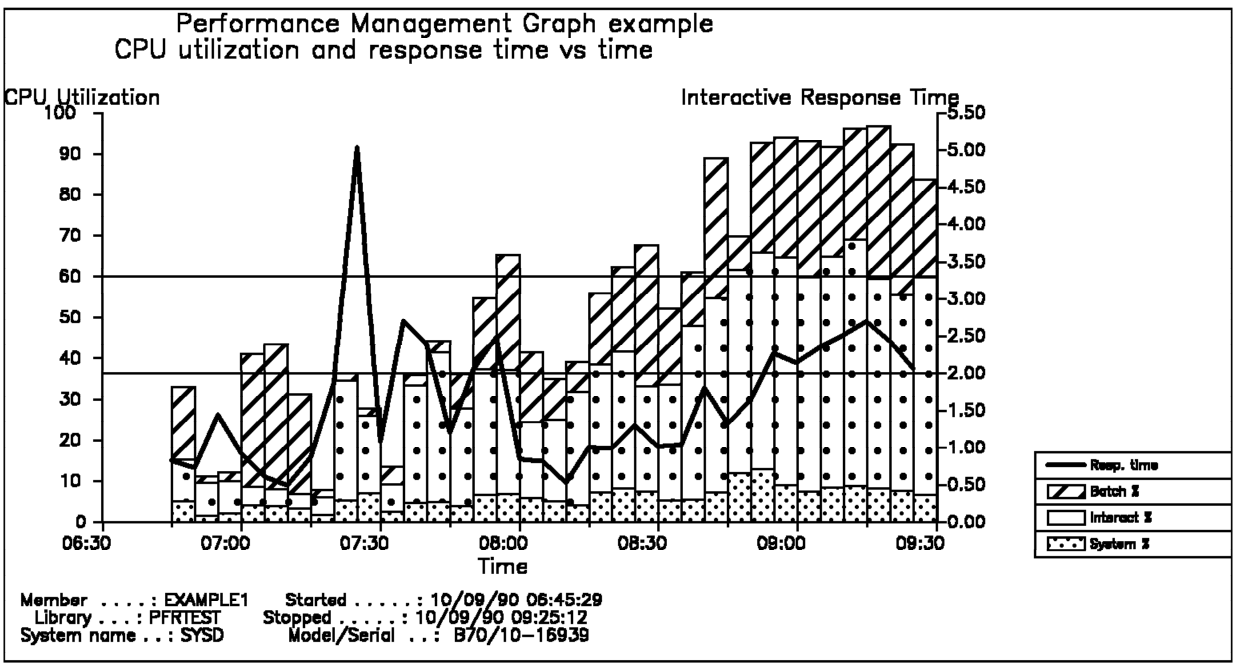


Figure 16. CPU by Type + Response Time. Profiles P01 to P33

The above graph represents all the records listed in the previous query, Figure 14 on page 77. It can be seen 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.

The following discusses other observations that can be made by querying and analyzing the QACPPROF records. A more detailed treatment of problem analysis is covered in Chapter 7, "Performance Problem Analysis" on page 91.

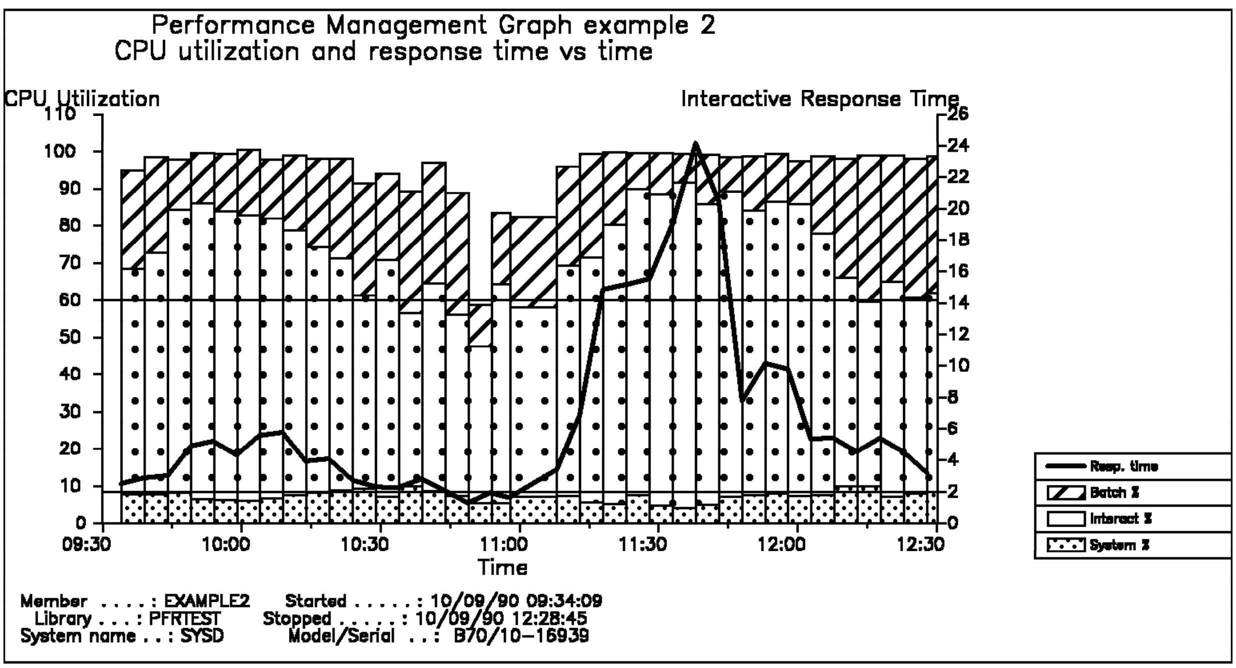


Figure 17. CPU by Type + Response Time. for Profiles P34 to P68

This graph shows the poor response time between 11:15 and 12:00 which coincides with the times on the query listing in Figure 15 on page 78, where the variable exceptional wait time was so high.

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.

6.7 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 command CRTHSTDTA on every measurement member you want included. Menus accessed via the Performance Tools STRPFRT command also provide this function. After this creation of historical data has completed, 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 would be prudent to save the data to suitable magnetic media.

The command DSPHSTGPH 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 month period. The data includes CPU utilization versus response time, but could 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.

Chapter 10 of *AS/400 Performance Tools Guide* provides more information on the use of performance graphics.

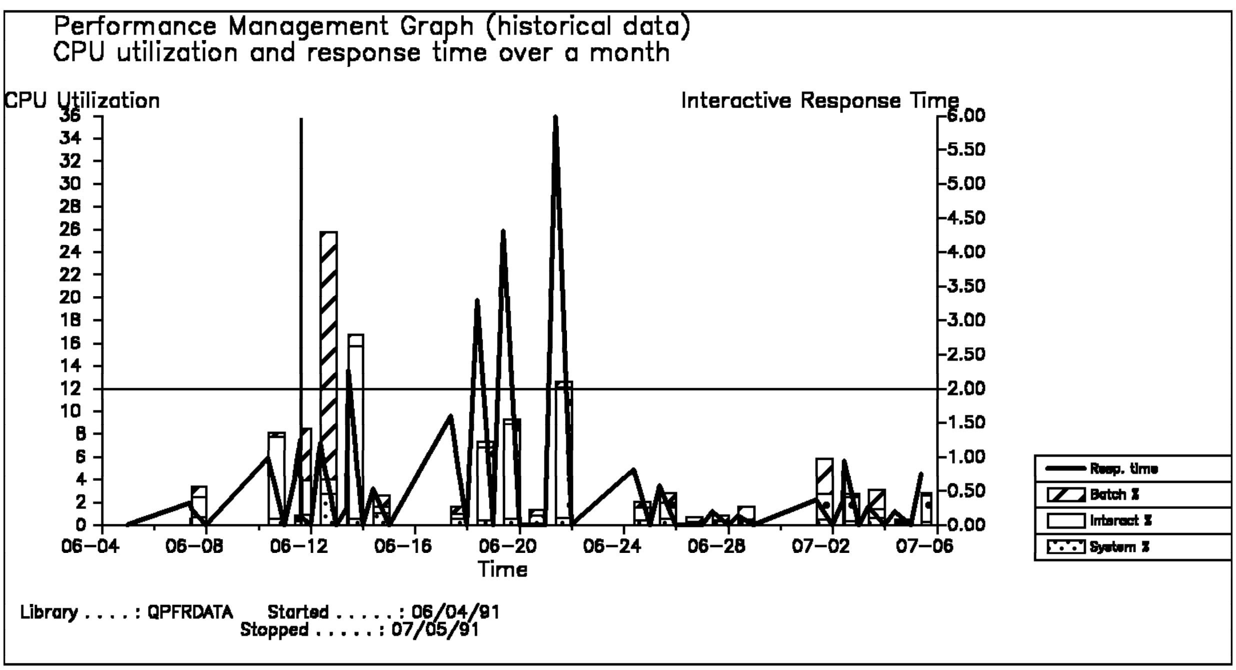


Figure 18. DSPHSTGPH Example Showing CPU Utilization vs Response Time

6.8 Archiving Performance Trend Data

A simple way to store the information is probably the way it gets into the trend analysis, in the form of a record in the measured profile file. Each measurement occupies just over one KB. After a few years there might be a few thousand records in the file but this should be no problem from a capacity point of view.

You have to consider that the measured profile data does not include transactions for pass-through or Client Access/400 5250 Emulation under RUMBA OS/2 Communications Manager, or Work Station Function (WSF) under V2R3 PC Support/400.

The BEST/1 Capacity Planning tool can build models using performance monitor data, and includes pass-through, Client Access/400 (PC Support/400), and 5250 emulation under RUMBA and Communications Manager. These models can be used to show a snapshot of performance history.

6.8.1 Historical Data

The information for the historical data is stored in two files and will take up more space than the profiles do.

- QAPGHSTD
- QAPGHSTI

The record lengths for these files are small (less than 150 bytes), but for each measurement there will be a few hundred records in both files. The total size (including an access path) might range from 0.5 to 2MB for a month of data.

6.9 Performance Management/400 (PM/400)

If PM/400 service is available in your country, this is the best way to store performance trend data and to obtain performance trend reports and excellent graphs. Depending on the specific contract agreement reports and graphs are made available monthly, quarterly, or annually.

PM/400 primarily processes Performance Monitor data and generates summary database files. This summarized data is then transmitted to a central site, normally operated by IBM. The summary data is analyzed and reports and graphs are generated.

When PM/400 is installed you do not have to depend on highly skilled personnel to analyze the Performance Monitor data. However, if detail analysis of the application environment is required, then a person skilled in Performance Monitor data analysis is required.

There is no known interface between PM400 data and the BEST/1 capacity planner. More information about this service can be found in 8.5, “Performance Management/400 (PM/400)” on page 119.

6.10 Performance Investigator

This tool is not intended to analyze trends in the medium or long term but it can help, if you are interested in analyzing performance trends in real time during the day.

More information about this tool can be found in 8.7, “Performance Investigator/400 PRPQ P84211 (5799-PRG)” on page 126.

6.11 BEST/1 Capacity Planning Tool

If you are interested in seeing how the changes to your configuration would affect performance, you can do so using the BEST/1 Capacity Planning Tool, which was introduced for V2R2. It replaced 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 *pre-defined workloads* or by explicit user entry of transaction definitions

Models created from actual performance monitor data offer the most accurate capacity planning results.

6.11.1 BEST/1 tips

The following is a list of various tips or limitations when modeling BEST/1.

- BEST/1 does not model 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 File Server I/O Processor (FSIOP) utilization is reported with other communication IOPs in the PerformanceTools/400 reports, BEST/1 does not model the FSIOP, such as varying cache memory sizes

  There is insufficient experience with LAN Server/400 and FSIOP environments to develop modeling algorithms. Since most of the ″work″ is done in the File Server I/O Processor, 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 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 only available on 6602, 6603, 9337-2xx and 9337-4xx disks. The I/O characteristics of the application will 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 will only have 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 could be 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 will either increase or decrease transaction rates which can then be 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 the user recorded that this number of transactions completed in 4 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 4 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 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 will decrease if other higher priority activity is increased, and vice versa.
  – Batch jobs cannot be running in a pool with interactive work.

- V3R1 supports 24 new pre-defined workloads, based upon PM/400 performance data.

These new workloads include 22 industry workloads and 2 client/server workloads that are described in Appendix B of the *BEST/1 Capacity Planning Tool Guide (SC41-3341)*

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.

  The new higher ″acceptable″ and ″poor″ page fault rates as documented in the *Work Management Guide - Version 3* and Appendix A, "Guidelines for Interpreting Performance Data" on page 323 are supported. The new Disk IOP Utilization guideline (70%) and threshold (80%) values documented in Appendix A, "Guidelines for Interpreting Performance Data" on page 323 are supported, though the BEST/1 manual still shows the older 45% and and 50% values for the disk IOP utilization.

- BEST/1 will now support DASD IOPs as follows:

  – 6502 Write Cache (High Availability)

  – B502 As above (only 8 device)

  – 6530 no Cache or High Availability

  – B530 As above (only 8 devices)

  – 9152,9153,9171,9172,9173 low end IOPs

- BEST/1 supports the new disk devices announced in 1994 as shown in Figure 19 on page 86.

```
======================================================================
 ARM TABLE SETTINGS - ONLY NEW VALUES SHOWN


    Drive         Capacity     <----------- milliseconds ---------->
    Feature         (MB)       Ctlr  Transfer   Seek   Latency Total
    -------------   ----       ----  --------   ----   ------- -----
    6602            1031        .1     .2        7.2     5.6    13.1
    6602-050 WC     1031        .7     .2        1.2     5.6     7.7
    6602-070 WC/R   1031        .7     .2        1.2     5.6     7.7
    6603            1967        .1     .2        7.8     5.6    13.7
    6603-050 WC     1967        .7     .2        1.2     5.6     7.7
    6603-070 WC/R   1967        .7     .2        1.2     5.6     7.7
    6605            1031        .1     .2        7.2     5.6    13.1
    6605-050 WC     1031        .6     .2        1.2     5.0     7.0
    6605-070 WC/R   1031        .6     .2        1.2     5.0     7.0
    6606            1967        .1     .2        6.4     5.0    11.7
    6606-050 WC     1967        .6     .2        1.2     5.0     7.0
    6606-070 WC/R   1967        .6     .2        1.2     5.0     7.0
    6607            4194        .1     .2        6.8     5.2    12.3
    6607-050 WC     4194        .6     .2        1.3     5.2     7.3
    6607-070 WC/R   4194        .6     .2        1.3     5.2     7.3
    9337-220 WC      970       1.6     .4        1.5     7.0    10.5
    9337-222 WC/R    970       1.7     .4        1.5     7.0    10.6
    9337-225 WC      970       1.6     .2        1.2     5.6     8.6
    9337-227 WC/R    970       1.7     .2        1.2     5.6     8.7
    9337-240 WC     1967       1.6     .2        1.2     5.6     8.6
    9337-242 WC/R   1967       1.7     .2        1.2     5.6     8.7
    9337-420 WC      970       1.7     .2        1.2     5.0     8.1
    9337-422 WC/R    970       2.2     .2        1.2     5.0     8.6
    9337-440 WC     1967       1.7     .2        1.2     5.0     8.1
    9337-442 WC/R   1967       2.2     .2        1.2     5.0     8.6
    9337-480 WC     4194       1.7     .2        1.3     5.2     8.4
    9337-482 WC/R   4194       2.2     .2        1.3     5.2     8.9

 WC = Write Cache is available
 WC/R = Write Cache and RAID-5 is available

The nn2 model designation cannot be ordered. It merely
conveys to BEST/1 that RAID-5 modeling is requested. The drives will
report as the order-capable 9337-xxx devices for all other
AS/400 functions.

The configuration rules for the drives are essentially the same
as the prior 9337 and 66xx technology.
```

Figure 19. BEST/1 Disk Model Specifications - New V3R1 Values Shown

The 9337CACHE device in the hardware table is now set to assume 75% write cache effectiveness and the 4MB write cache option available for 6501 attached disks. This value is used when doing 'what-if' analysis from other DASD technologies up to 9337 technologies.

New 660x devices include the following:

- 6605 (1031MB), 6606 (1967MB), and 6607(4194MB)
- 6605-050 (1031MB), 6606-050 (1967MB), and 6607-050 (4194MB) which attach to the 6502 and 6522 DASD IOPs with 2MB write cache support.
- 6605-070 (1031MB), 6606-070 (1967MB), and 6607-070 (4194MB) which attach to the 6502 and 6522 DASD IOPs with 2MB write cache support and the RAID-5 high availability option active.

The V2R3 and V3R0M5 'Edit Sync Reads Guidelines' panels have been updated. V2R2 does not include this panel but has been updated internally to reflect the new memory guidelines.

```
                                   -----All Other Pools------
CPU relative performance           Guideline        Threshold
2.0 or less  . . . . . . . . . . . .   15               25
3.0 or less  . . . . . . . . . . . .   25               40
10.0 or less . . . . . . . . . . . .   35               60
30.0 or less . . . . . . . . . . . .   80               130
50.0 or less . . . . . . . . . . . .   180              300
100.0 or less  . . . . . . . . . . .   250              440
Greater than 100.0 . . . . . . . . .   300              500
```

6.12 Manual Batch Run Time Estimation

Batch run time modeling is possible under BEST/1 as discussed under 6.11.1, "BEST/1 tips" on page 83. Other batch run time modeling tools are discussed in Appendix C, "IBM Internal Use Only Tools/Documents" on page 409. This section presents a formula for manually predicting run time, based on Performance Tools/400 reports and relative CPU RIP ratings and disk service times.

For relative CPU processor speed RIP ratings use Table 64 on page 346 and Table 68 on page 348.

For disk service times use Table 73 on page 353.

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 data base or other object contention. Neither does it include the impact of other jobs running concurrently on the system and their job priorities relative the 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 formula below).
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 formula below).

Formulas:

CPU Queuing Calculation:

$$\text{CPU QM:} \quad \frac{1}{\left[1 - \left(U1 + \left(U2 \times \frac{N-1}{N} \right) \right) \right]^{P}}$$

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 =

$$100 \times \frac{(\text{Physical Disk OPS per second} \times \text{Base Disk Service Time})}{\text{Number of Disk Arms}}$$

Physical Disk OPS per Second: Measured with Performance Tools using Disk Utilization section of the System Report.

DISKRESP = Base Disk Service Time x DISKQM

While Table 73 on page 353 provides average disk service times, you may also review the System Report Disk Utilization section heading "Average Time per I/O" to see the DISKRESP for a specific application environment. If you run the batch job on the current system, according to your desired concurrency with with other jobs, we recommend you bypass the calculation above and use the average "Response" time of all your busy disks from the report for DISKRESP.

6.13 Comparing Extrapolated Data with Historical Data

6.13.1 Example of Trend Analysis on Response Time

In the following example we show how collected measured profile data can be summarized in monthly averages and compared with the current month data.

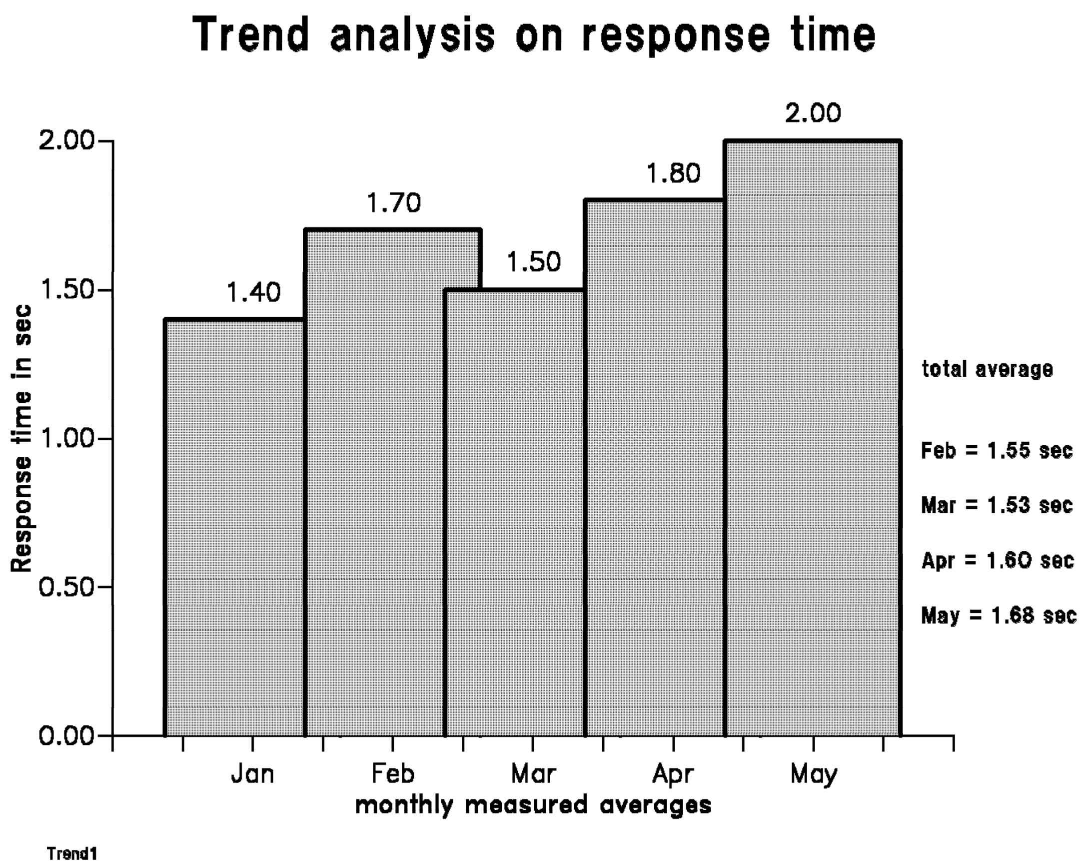


Figure 20. Trend Analysis Example on Response Time

In the example above a person can easily detect the trend upwards in response time. However, the program for trend analysis in the Automated Performance Management Tool looks at these values in a little broader view. It creates averages for the last month and compares this average with all previous month averages.

You could use two comparisons: the current month compared with the previous or the current month with all the previous months. We chose to use all the previous months′ values to compare with the current month. This approach accommodates some fluctuation every month without overreacting to a single instance of a large change between months.

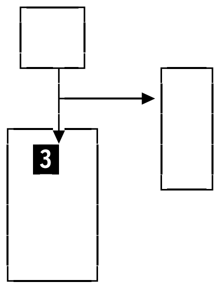
6.14 Notifications of Performance Trends

It is important to advise management of the results of 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 (a) making no changes to the system, and (b) installing recommended upgrades using the facilities of BEST/1 Capacity Planning Tool.

Chapter 7. Performance Problem Analysis

7.1 Contents

This chapter addresses stage **3** of the performance management flowchart block of activity shown in Chapter 4, “Performance Management Methodology” - Performance Problem Analysis.



This chapter provides assistance in 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 very familiar with the contents of Chapter 4, “Performance Management Methodology,” Appendix A, “Guidelines for Interpreting Performance Data,” Chapter 9, “System Performance Tuning Tips” and Chapter 10, “Design and Coding Tips.”

Appendix B, “Field Descriptions and Sample Performance Reports” should be used in conjunction with the *AS/400 Performance Tools Guide* when interpreting the output of various Performance Tools reports. This appendix information is provided as a supplement to the *Performance Tools Guide* information.

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.

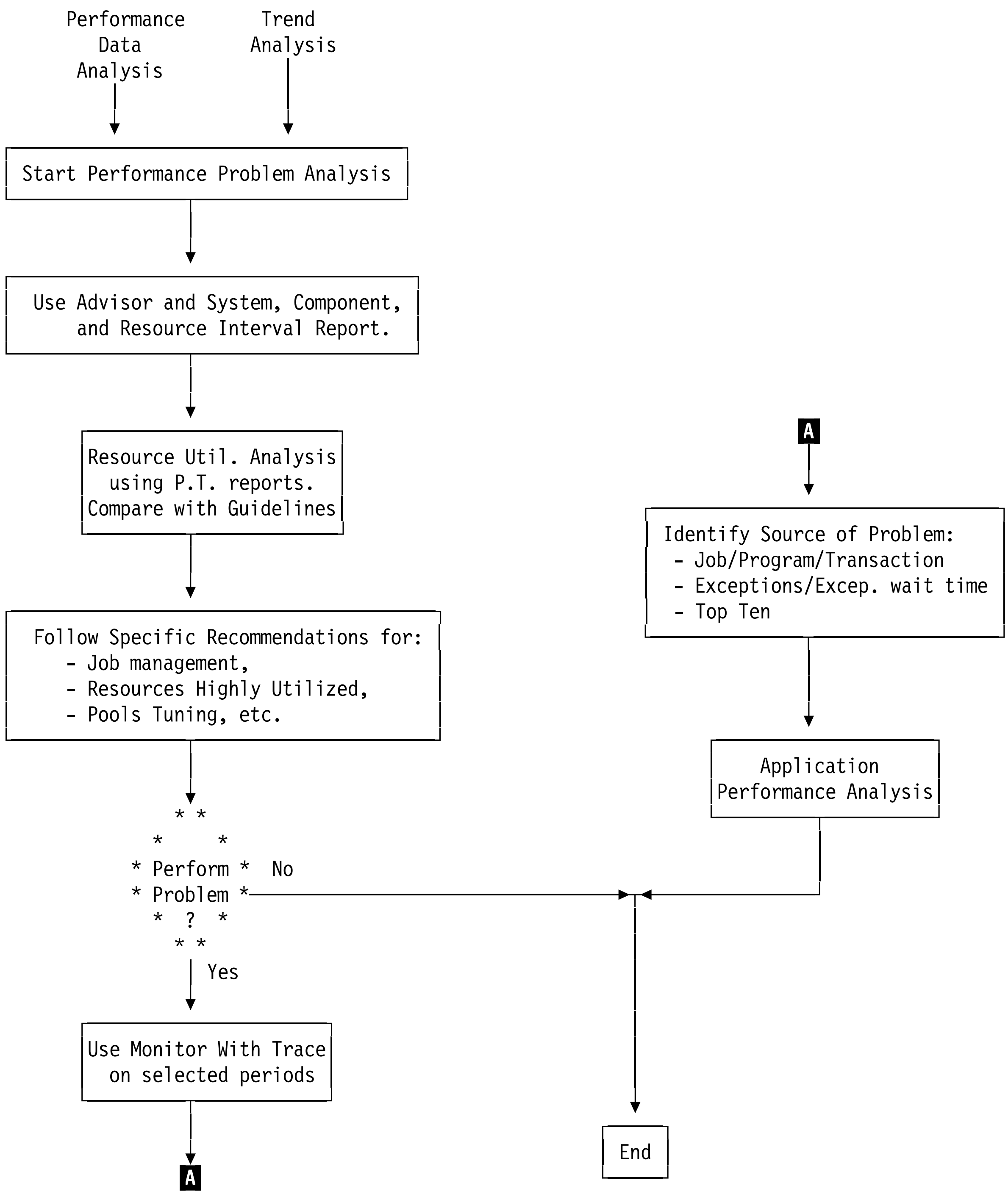


Figure 21. Performance Problem Analysis Flowchart

7.2 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 simple like 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 which 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.

7.3 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 very 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), 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, etc.

Sometimes a single job can cause severe degradation on many other jobs. This could be a CPU-intensive job that is running at a very 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.

7.4 Averages Versus Peak Workloads

Is the performance always unacceptable or do the users notice degradation regularly? For example, are there problems before/after lunch break or at the end of a working day? If performance is always unacceptable you should review Chapter 9, “System Performance Tuning Tips” on page 131 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 techniques discussed in 7.5.1, “Interactive Analysis” on page 94.

7.5 Reviewing the Measured Data

7.5.1 Interactive Analysis

- 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 disk in the system and the number of addresses used. By selecting assistance level 3 (Advanced) you can monitor all 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 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 Figure 31 on page 154.

- 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 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 via 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. User the Performance Tools/400 reports and BEST/1 for capacity planning.

  You should compare values on this display to Table 24 on page 323 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 via 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 the other commands discussed here.

You should use the automatic refresh with refresh rate of 5 seconds and see if the same job or task surfaces in many intervals. 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 etc. Find out what application is used and try to recreate 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 7.8, "Application Level Problem Analysis" on page 105 for more information.

- DSPPFRDTA

  This command is a part of the *Performance Tools/400* (5738-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 have been issued **before** you can analyze the performance data.

  – **Summary data**

    To analyze system wide data with the DSPPFRDTA command, you start the data collection with value *NONE of the parameter TRACE.

  – Refer to *Performance Tools/400 Guide*, SC41-8084 for more information about the DSPPFRDTA command.

7.6 System Wide Performance

The ways to analyze system wide performance are:

- Interactively, for example with the WRKSYSSTS command and
- Using *Performance Tools/400* reports or the Display Performance Data (DSPPFRDTA) command

7.6.1 Memory Usage

7.6.1.1 Memory Performance Displays/Reports

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 25, Table 26 and Table 27 in Appendix A, "Guidelines for Interpreting Performance Data" on page 323.

DISPLAY PERFORMANCE DATA: The DSPPFRDTA command provides an interactive display of data given in the System, Component, and Interval reports.

DSPACCGRP/ANZACCGRP: Shows job temporary storage usage for one job or a group of jobs, open files, file I/O counts and active programs.

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.

7.6.1.2 Information about Memory Usage

Table 2. Memory Utilization		
Resource Description	**Where to Look**	**Compare With**
Machine pool NDB page fault	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Table 25 on page 323 Appendix A, "Guidelines for Interpreting Performance Data"
Sum of DB and NDB Page Faults for Each Pool	System Report: Storage Pool Utilization, ADVISOR	Table 26 on page 324 Appendix A, "Guidelines for Interpreting Performance Data"
Sum of DB and NDB Page Faults in All Pools	System Report: Storage Pool Utilization, ADVISOR	Table 27 on page 324 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	

7.6.2 CPU Performance

7.6.2.1 CPU Performance Reports/Displays

WRKACTJOB: This command allows you to determine:

- What is the utilization percentage of CPU?
- How much does each job use CPU, 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 would be of help. See Table 24 on page 323 for more information. Please remember that you should not use measurement times that are either under 5 minutes or over 30 minutes long.

7.6.2.2 Information about CPU Usage

Table 3 (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 24 on page 323 Appendix A, "Guidelines for Interpreting Performance Data"
CPU seconds per transaction	System Report: Resource Utilization, Transaction Report: Job Summary	Table 64 on page 346 Appendix A, "Guidelines for Interpreting Performance Data"

Table 3 (Page 2 of 2). CPU Utilization, Seconds		
Resource Description	**Where to Look**	**Compare With**
CPU Queuing Multiplier	Transaction Report: Job Summary, System Summary Data, System Report	Table 69 on page 348 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 4. Exceptions		
Resource Description	**Where to Look**	**Compare With**
Authority Lookup	Component Report: Exception Occurrence Summary	Table 34 on page 326 Appendix A, "Guidelines for Interpreting Performance Data"
Exceptional Address Overflow (EAO)	Component Report: Exception Occurrence Summary	Table 40 on page 332 Appendix A, "Guidelines for Interpreting Performance Data"
Size (Arithmetic Overflow and Binary Overflow)	Component Report: Exception Occurrence Summary	Table 46 on page 337 Appendix A, "Guidelines for Interpreting Performance Data"
Verify	Component Report: Exception Occurrence Summary	Table 52 on page 340 Appendix A, "Guidelines for Interpreting Performance Data"
Decimal Data	Component Report: Exception Occurrence Summary	Table 58 on page 343 Appendix A, "Guidelines for Interpreting Performance Data"

7.6.3 Disk Performance

7.6.3.1 Disk Performance Reports/Displays

System Report

- Shows disk I/O by job type (Batch, System, Interactive, Pass-through, etc.)
- 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.

7.6.3.2 Information about Disk Performance

Table 5 (Page 1 of 2). Disk Utilization

Resource Description	Where to Look	Compare With
Disk Arm Utilization	System Report: Disk Utilization, WRKDSKSTS	Table 24 on page 323 Appendix A, "Guidelines for Interpreting Performance Data"

Table 5 (Page 2 of 2). Disk Utilization

Resource Description	Where to Look	Compare With
Disk IOP Utilization	Component Report: IOP Utilization	Table 24 on page 323 Appendix A, "Guidelines for Interpreting Performance Data"
Disk Physical I/O per Transaction (Average)	System Report: Resource Utilization, Transaction Report	Table 72 on page 352 Appendix A, "Guidelines for Interpreting Performance Data"
Disk Physical I/O per Transaction per Job.	Transaction Report: Job Summary	Table 72 on page 352 Appendix A, "Guidelines for Interpreting Performance Data"
Synchronous and Asynchronous DB and NDB I/O per Job by Interval	Job Interval Report	
Synch. and Asynch. 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: Data Base Journal Summary	
Synch. and Asynch. Disk I/O by Subsystems and Pools by Interval	Pool Report: Subsystem activity	

Note:

Use the Write, Read, and Total Physical Disk I/O per transaction values shown in Appendix A as a ″reasonability measure.″ Verify any job exceeding the values is performing the work required.

Note that 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 the index entry for the QDBSRVnn jobs for more information.

7.6.3.3 Other Useful Reports

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 and Checksum Set), 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.

7.6.4 Local Workstation IOP and Communications Lines

7.6.4.1 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 could occur if there were heavy use of the text assist functions for an OV/400 editor. High twinaxial utilization could occur if there were a significant amount of high-speed printer output or PC Support/400 shared folder or file transfer work going on.

7.6.4.2 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 the Performance Tools.

Note the Performance Tools also provide 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 will be needed in most cases.

7.6.4.3 Information about Local Workstation and Communications Performance

Table 6 (Page 1 of 2). Lines and IOPs Utilization

Resource Description	Where to Look	Compare With
Local WS IOP	Component Report: IOP Utilization	Table 24 on page 323 Appendix A, "Guidelines for Interpreting Performance Data"

Table 6 (Page 2 of 2). Lines and IOPs Utilization		
Resource Description	**Where to Look**	**Compare With**
Multifunction IOP	Component Report: IOP Utilization	Table 24 on page 323 Appendix A, “Guidelines for Interpreting Performance Data”
Communication IOP	Component Report: IOP Utilization	Table 24 on page 323 Appendix A, “Guidelines for Interpreting Performance Data”
File Server IOP	Component Report: IOP Utilization	IOP reported is the one for exchanging data between FSIOP and AS/400 Disk. No guideline available at this time. Attached LAN lines are reported under remote lines, LAN lines. See index entry *FSIOP performance queries* for information on cache read/write hit and 486 CPU utilization percentage guidelines.
Remote Lines, LAN Lines	System Report: Communication Summary, Resource Report	Table 24 on page 323 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 Jobtype	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 Work Stations - Response Time Buckets	
Local and Remote Workstation Response Time Distribution By Interval	Resource Report: Local Workstation IOP Utilization and Remote Workstation Response Times	

7.6.4.4 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, etc.

  See 9.13, “Communication Performance Considerations” on page 168 for information on the OS/400 Peformance 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.

7.6.5 Activity Level Performance Data

Performance Monitor, WRKSYSSTS: The OS/400 Performance Monitor trace and sample data, and WRKSYSSTS provide information on activity level usage and job state changes.

7.6.6 Activity Level Performance Reports/Displays

System Report: Job state changes (movement in and out of activity level) per pool for the total collection period.

7.6.7 Comparing with Activity Level Guidelines

Table 7. Activity Level

Resource Description	Where to Look	Compare With
Activity Level for *BASE and Spool Writer pool	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Fig. 3-15, 3-16 and 3-17 in Performance Tools Guide page 3-17
QINTER Activity Level.	System Report: Storage Pool Utilization, WRKSYSSTS, ADVISOR	Fig. 3-18 in Performance Tools Guide pag. 3-18

7.6.8 W – I and A – W Ratio Guidelines

More simply, get the activity level to where W – I is a little bit above zero and then increase the activity level by 2.

If W – I is always zero, the activity level is too high.

7.6.9 Comparing W-I and A-W Ratio Guidelines

Table 8. W-I and A-W Ratio		
Resource Description	**Where to Look**	**Compare With**
W-I/A-W	System Report: Storage Pool Utilization, WRKSYSSTS	Table 28 on page 324 Appendix A, "Guidelines for Interpreting Performance Data"

7.6.10 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 via 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.

7.6.11 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
- System impact of authority lookups
- System impact of exceptions (such as numeric overflow and Effective Address Overflow (EAO))
- Communication line utilization and error percentages.

The Advisor output is grouped under the following headings: Recommendations, Conclusions and Interval Conclusions.

The Advisor will suggest 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 could 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 would not be made if the data was collected on another system.

The Advisor also tells you what report you should run to get more information for your problem analysis. See the ″Advisor″ chapter of the *Performance Tools Guide* for detailed information.

7.6.12 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 STRPFRMON RRSPTIME value other than the default *NONE to have the response time data collected from the active 5494 controllers. A value of *SYS will use the same response time slots (0-1 seconds, etc.) as for the LRSPTIME (local response time) parameter.

7.7 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 7.5.1, “Interactive Analysis” on page 94.

User level problem analysis is done:

- By WRKACTJOB command
- By WRKSYSACT command
- By DSPPFRDTA command
- By analyzing Performance Tools/400 reports

7.7.1 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 ″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 performance problem? If all the users of this program have problems with performance you should see 7.8, “Application Level Problem Analysis” on page 105 for more information.

Refer to ″System Summary Data″ section, ″Analysis by interactive Response Time″ to see how your response time objectives are met.

7.7.2 Print Transaction Summary Report

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 Transaction Detail report and 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.

7.8 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 ″Interactive program statistics″ section of ″Job Report″ for the top 10 programs with highest resource utilization like:

- 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 with Table 64 and Table 72 in Appendix A, “Guidelines for Interpreting Performance Data” on page 323 to see if values are acceptable.

Please note that values shown are guidelines only so you must verify that each transaction exceeding the values is performing the work required.

7.8.1 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 under 7.8.2.1, “Programmer Performance Utilities” to find out the cause of the problem.

7.8.2 Useful Tools for Performance Analysis

7.8.2.1 Programmer Performance Utilities

Job Trace: OS/400 commands STRSRVJOB, ENDSRVJOB and TRCJOB can be used to produce trace job information. The Performance Tools STRJOBTRC and ENDJOBTRC, and TPST′s PRTTRCSUM commands can also be used to produce trace job information.

The PRTTRCSUM command from TPST is a very productive tool for finding problems.

The trace job output can be used to determine the following information that can be used 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 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.

Caution, for a job with hundreds of user program/procedure calls, trace job may have a significant impact on CPU utilization.

Sampled Address Monitor (SAM): The Sampled Address Monitor commands are used to find ″hot spots″ within a program once the program using lots of CPU time has been identified. They are normally only used when problems with a program are suspected. The SAM functions can be used in the normal production environment with no modification required to the program under examination.

The sequence of ″SAM″ commands is:

1. STRSAM (Start the Address Monitor)

   You can specify the program names to collect data for and, optionally ILE information such as module name and procedure name.

2. STRSAMCOL (Start the actual collection of address data)
3. ENDSAMCOL (End the collection of data)

   Output is directed to database files QAPTSAMH and QAPTSAMV, which are input to the PRTDAMDTA command processing.

4. ENDSAM (Ends the STRSAM collection parameters and deletes internal SAM data)

   You should end SAM as soon as possible as it degrades system performance.

5. PRTSAMDTA (print the collected address data)

   Output includes a histogram giving CPU time per individual HLL instruction in a program.

STRDSKCOL and PRTDSKRPT: The report shows the database and other object activity on a number of different drives. If you had a drive with high usage and some database file(s) could be identified that had high activity, the file(s) could be copied to other drives.

If PRTSYSRPT or PRTCPTRPT reports show some disk devices consistently have higher activity than others, you may be able to determine what objects on the device have the most activity using these commands.

- If one file is particularly active and only a few records are in use, the application should be investigated to see what can be done with it.
- If one file is very active and many records are in use, reallocating, copying, and deleting the file (using the CL CPYF command) may reduce the device activity for the device. Note, you do not use Save/Restore for these operations.
- If multiple files have consistently high activity on a specific device, moving one or more of them (using CPYF) to another device may reduce the activity for the device.

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
4. The active programs within the jobs selected

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-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 as ILE C/400 was introduced in this release.

The biggest changes are that 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 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 V3R1 the Work Control Block 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 heep storage areas.

7.8.3 Print Transaction Detail Report

If you need a more detailed problem analysis you 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 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 will surely be unacceptable.

7.8.4 Print Transition Report

If you want to know all 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 will 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 Appendix B, “Field Descriptions and Sample Performance Reports” on page 355.

- 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, etc.

Since these functions have a significant impact on system and job performance you should work with the customer to reduce their usage.

You may also see in the Transaction report the I/O Transaction Boundaries. 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 *AS/400 Programming: Performance Tools/400 Guide*, SC41-8084, for more details.

Note that through March 1995 V3R1, the SOT, EOT, etc. ″eye catchers″ do not appear in the Transition Detail reports. A PTF to the system was not available at the time of redbook publication.

7.8.5 Timing and Paging Statistics (TPST) PRPQ

While the Print Transaction Report (PRTTNSRPT) command can be used to determine what programs are using the majority of the CPU time, the algorithm that identifies the program using the CPU is not accurate in all cases. Use of the TPST PRPQ is preferable.

The AS/400 Timing and Paging Statistics (TPST) PRPQ helps identify programs with excessive CPU and disk activity and how many times each program is called. Once the high use programs are identified, the Performance Tools Sampled Address Monitor (SAM) commands can be used to identify which program instructions are using the most CPU time.

For more information about the usage of TPST PRPQ see Chapter 8, “Additional Performance Tools” on page 111.

Chapter 8. Additional Performance Tools

This chapter contains a list of tools useful for performance management in addition to the capabilities of the Performance Tools, licensed program 5738-PT1.

8.1 Timing and Paging Statistics (TPST) PRPQ

TPST is very useful for determining the CPU time for a specific application function, and for determining what proportion of the CPU time is attributable to each program in the function. In addition, TPST lists the number of times a program is called and allows you to compare the data in different TPST measurements. It can be used to evaluate coding alternatives and the effect each has on performance. Furthermore it allows you to focus your application tuning efforts on the worst performing programs or the most called modules.

Version 2 Release 3 TPST is identified as P84204 (5799-EER) and Version 3 Release 0 TPST is identified as P84257 (5799-FQA). Their capabilities are identical as ILE support (optionally including ILE module and procedure names) was introduced with V2R3 ILE C/400.

TPST is a program-oriented performance tool, not a job-oriented tool. With TPST there is no way you can link specific program usage to a specific job. For this reason best results are obtained when only the application being analyzed is running in the system. In this controlled environment you know that all system modules are called by the application.

The Performance Tools - Transaction report charges the calling program with all the resources used by any programs called by it but which do not interact with the workstation. This may result in an inaccurate representation of programs that are high resource users. An example of this is the QUIINMGR (User Interface Manager Interaction Program) which is the ″menu manager″ for IBM-supplied menus. This program can be charged as a long transaction if a program called from the command screen displayed by QUIINMGR performs excessive processing but never issues a Write to the screen. Upon return to QUIINMGR, QUINMGR writes to the screen and gets charged with the long transaction since it issues the first write to the screen following input from the workstation.

Examples that could assign poor performance to QUIINMGR include:

- SNDNETF for a large save file

  SNDNETF copies the save file to an internal space for a SNADS send job to distribute

- SNDNETSPLF for a large spooled file

  SNDNETSPL copies the printed output to an internal space for a SNADS send job to distribute

- A long running query that produces printed output only
- A program that builds a large database file

TPST is able to capture precise data on each program′s resource usage by interrogating resource counters at each program-to-program transition. The ENTTPST command establishes an operating environment for the TPST monitor.

Specific programs can then be monitored using the STRTPST command.You should end TPST as soon as possible as it may add significant CPU utilization on a busy system.

The TPST Print Trace Summary (PRTTRCSUM) command can be used to process the Performance Tools database file data optionally produced by the OS/400 TRCJOB command OUTFILE parameter or the Performance Tools STRTRCJOB/ENDTRCJOB command output file QAPTTRCJ. PRTTRCSUM output can be used to determine the frequency of calls to user programs, HLL initialization routines, and file open and close routines. Exception information, such as Effective Address Overflow (EAO) and Size data can be shown for each program.

Use TPST when problem analysis has identified inexplicable CPU or disk activity within a job or system task. TPST produces program/task statistics, but does not identify the jobs using these programs. Either the job trace functions or the Performance Tools Transaction report output can identify programs used within a job.

TPST also provides the function called Compare TPST (CMPTPST) which can be used to compare the data in different TPST measurements. It can be used to evaluate coding alternatives and the effect each has on performance. The purpose of CMPTPST is to allow the user to get a tabular and graphical view of the relationship of up to 9 TPST measurements in one report. This work can be done by comparing the results of the LSTTSPT command as shown in 8.1.1, “Comparing Coding Techniques: An Example,” but this is a tedious process if you need to compare many measurements.

8.1.1 Comparing Coding Techniques: An Example

To show how TPST can help you in your environment we wrote two programs and we analyzed them with TPST. OPENSHRNO is a program that calls another program, called OPEN10, 30 times which in turn opens 10 different files, producing 300 full opens. OPENSHRYES overrides the files with SHARE(*YES) and calls OPEN10 30 times, producing 10 full opens and 300 shared opens. We analyzed both applications in an almost dedicated environment with TPST and we printed the reports sorted by cumulative CPU time; that is, the CPU time used by the program and all the programs invoked by the program directly or indirectly.

TPST - SORTED BY CCPU 08/04/92 13:37:41 PAGE 8

OPEN with SHARE(*NO)

STANDARD REPORT
MODULE SECTION

NAME		INVOKED	CPU MILS/ % /RANK	SYNC I/O	ASYNC I/O	MI CALLS OR XCTLS	MI SVLS	CUMULATIVE MILS/ % /RANK		CUMULATIVE SYNC IO	ASYNC IO
QUOCMD		1	7/ 0.0/044	0	0	4	1	9014/91.3/001		2	0
OPENSHRNO	1	1	16/ 0.1/036	0	0	32	1	8976/90.9/002	A	2	0
OPEN10	2	30	308/ 3.1/008	0	0	660	30	8464/85.7/003		0	0
QDMCOPEN	3	300	792/ 8.0/004	0	0	300	1500	4396/44.5/004		0	0
QDMCLOSE	4	300	416/ 4.2/006	0	0	300	1200	3211/32.5/005		0	0
*DESCR	5	300	2547/25.8/001	0	0	0	0	2547/25.8/006		0	0
*CRTDOBJ		300	1388/14.0/002	0	0	0	0	1388/14.0/007		0	0
*RSLVSP		893	1373/13.9/003	0	0	0	0	1373/13.9/008		0	0
QDBOPEN	6	300	356/ 3.6/007	0	0	0	600	1004/10.1/009		0	0
*ACTCR		300	596/ 6.0/005	0	0	0	0	596/ 6.0/010		0	0
QLRMAIN		60	121/ 1.2/010	0	0	60	120	544/ 5.5/011		0	0
QCLCLCPR		30	103/ 1.0/014	0	0	60	60	481/ 4.8/012		2	0
QUYLIST		4	46/ 0.4/026	0	0	68	8	430/ 4.3/013		0	0
QWCCRCRC		30	74/ 0.7/021	0	0	90	30	358/ 3.6/014		0	0
QWSPUT		57	206/ 2.0/009	0	0	25	67	293/ 2.9/015		0	0
QMHRTMSS		39	84/ 0.8/018	0	0	40	77	223/ 2.2/016		0	0
QPERCLSP		30	83/ 0.8/019	0	0	0	150	221/ 2.2/017		0	0
QMHSNJMQ		35	93/ 0.9/015	2	0	2	326	147/ 1.4/018		2	0
QMHMFSRH		42	61/ 0.6/024	0	0	0	126	131/ 1.3/019		0	0
*MATPTR		724	120/ 1.2/011	0	0	0	0	120/ 1.2/020		0	0
*LOCK		303	107/ 1.0/012	0	0	0	0	107/ 1.0/021		0	0
**UNKNOWN		0	106/ 1.0/013	0	0	0	0	106/ 1.0/022		0	0
*UNLOCK		303	93/ 0.9/015	0	0	0	0	93/ 0.9/023		0	0
QSFPUT		32	89/ 0.9/017	0	0	0	32	90/ 0.9/024		0	0
**LICTSKS		0	82/ 0.8/020	0	0	0	0	82/ 0.8/025		0	0
*MATPRATR		393	72/ 0.7/022	0	0	0	0	72/ 0.7/026		0	0
QDBCLOSE		300	66/ 0.6/023	0	0	0	0	66/ 0.6/027		0	0
QWSGET		14	33/ 0.3/028	0	0	4	22	55/ 0.5/028		0	0
QMHRCMSS		12	30/ 0.3/030	0	0	2	68	54/ 0.5/029		0	0
QWSSFLCT		20	49/ 0.4/025	0	0	0	20	51/ 0.5/030		0	0
QLMEVTHD		1	1/ 0.0/068	0	0	1	1	49/ 0.4/031		3	3
QLMCRSHD		1	2/ 0.0/058	0	0	4	2	48/ 0.4/032		3	3
QDMRCLSE		30	25/ 0.2/033	0	0	60	0	45/ 0.4/033		0	0
.		.	.	.	.	.	.	.		.	.
.		.	.	.	.	.	.	.		.	.
.		.	.	.	.	.	.	.		.	.

*Figure 22. Standard TPST Report for SHARE(*NO)*

In the *Timing and Paging Statistics Tool User's Guide* you can find an appendix with most of the OS/400 modules and machine interface names. For example QDBOPEN performs a full open, *DESCR a full close, and QDBSOPEN a shared open.

Looking at the *INVOKED* column for 3 to 6 in Figure 22 we can see that no other program in the system was opening database files. The CPU utilization for the program is in the second column. Total CPU used by program OPENSHRNO is 8976 as shown in A.

TPST - SORTED BY CCPU 08/04/92 13:35:12 PAGE 8

OPEN with SHARE(*YES)

STANDARD REPORT
MODULE SECTION

NAME	INVOKED	CPU MILS/ % /RANK	SYNC I/O	ASYNC I/O	MI CALLS OR XCTLS	MI SVLS	CUMULATIVE MILS/ % /RANK	CUMULATIVE SYNC IO	ASYNC IO
QUOCMD	1	9/ 0.1/052	0	0	4	1	3547/74.6/001	3	0
OPENSHRYES	1	41/ 0.8/030	0	0	62	1	3503/73.7/002 **B**	3	0
OPEN10	30	314/ 6.6/003	0	0	660	30	1926/40.5/003	0	0
QDMCOPEN	310	489/10.2/001	0	0	630	60	1000/21.0/004	0	0
QUYLIST	7	74/ 1.5/022	0	0	119	14	752/15.8/005	0	0
QMHSNJMQ	96	267/ 5.6/005	3	0	33	920	592/12.4/006	3	0
QLRMAIN	60	132/ 2.7/010	0	0	60	120	547/11.5/007	0	0
QCLCLCPR	30	98/ 2.0/014	0	0	60	60	532/11.1/008	2	0
QWSPUT	93	314/ 6.6/003	0	0	40	109	459/ 9.6/009	0	0
QCADRV	32	56/ 1.1/025	0	0	158	31	435/ 9.1/010	1	0
*RSLVSP	366	389/ 8.1/002	0	0	0	0	389/ 8.1/011	0	0
QDMCLOSE	310	216/ 4.5/006	0	0	10	340	365/ 7.6/012	0	0
QWCCRCRC	30	73/ 1.5/023	0	0	90	30	359/ 7.5/013	0	0
QMHMFSRH	72	207/ 4.3/007	0	0	0	216	324/ 6.8/014	0	0
QDBOPENC	10	10/ 0.2/048	0	0	20	10	294/ 6.1/015	0	0
QMHRTMSS	39	99/ 2.0/012	0	0	39	78	277/ 5.8/016	0	0
QDBSOPEN	300	92/ 1.9/016	0	0	10	10	222/ 4.6/017	0	0
QPERCLSP	30	90/ 1.8/018	0	0	0	150	220/ 4.6/018	0	0
QDBCLOSC	10	10/ 0.2/048	0	0	20	10	216/ 4.5/019	0	0
QSFPUT	53	142/ 2.9/008	0	0	0	53	145/ 3.0/020	0	0
**UNKNOWN	0	138/ 2.9/009	0	0	0	0	138/ 2.9/021	0	0
QDMGETOV	310	112/ 2.3/011	0	0	0	0	112/ 2.3/022	0	0
QWSGET	23	65/ 1.3/024	0	0	7	37	101/ 2.1/023	0	0
**LICTSKS	0	99/ 2.0/012	0	0	0	0	99/ 2.0/024	0	0
*MATPRATR	536	93/ 1.9/015	0	0	0	0	93/ 1.9/025	0	0
*DESCR	10	92/ 1.9/016	0	0	0	0	92/ 1.9/026	0	0
QWSSFLCT	32	86/ 1.8/019	0	0	0	32	89/ 1.8/027	0	0
QCAPOS	32	80/ 1.6/020	0	0	0	0	80/ 1.6/028	0	0
QCAFLD	32	78/ 1.6/021	0	0	0	2	79/ 1.6/029	0	0
QMHRCMSS	18	39/ 0.8/031	0	0	2	110	76/ 1.5/030	0	0
QDBOPEN	10	19/ 0.3/041	0	0	0	30	68/ 1.4/031	0	0
*LOCKSL	230	56/ 1.1/025	0	0	0	0	56/ 1.1/032	0	0
QCARULE	32	19/ 0.3/041	0	0	0	64	52/ 1.0/033	0	0
*FNDINXEN	72	48/ 1.0/027	0	0	0	0	48/ 1.0/034	0	0
QDMRCLSE	30	31/ 0.6/038	0	0	60	0	48/ 1.0/034	0	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

*Figure 23. Standard TPST Report for SHARE(*YES)*

Figure 23 shows the standard report for the program that performs shared opens instead of full opens. Comparing Cumulative CPU Utilization in Figure 22 on page 113 (**A** and **B**) we can conclude that changing this program to share open data paths resulted in an almost threefold decrease in CPU time.

Figure 24 on page 115 shows an example of TPTST module level output that includes ILE programs as well as OPM programs. Note that **1** shows that TPST does not record disk I/O for ILE modules. At time of redbook publication this is considered a permanent restriction. The disk I/O information was discontinued for ILE programs/modules to reduce TPST CPU impact in environments where hundreds of ILE calls are made within a few seconds.

```
                                                     TPST - SORTED BY CCPU        3/06/95  17:06:29 PAGE 17
                                                  BUPINZF, BUPAP1RN, ILECLVL2
                                                       COMP OPM to ILE info
                                             CUMULATIVE MODULE I/O BREAKDOWN REPORT
                                 SYNC READS           SYNC WRITES          ASYNC READS          ASYNC WRITES
NAME          INVOKED       DB         NDB         DB         NDB         DB         NDB         DB         NDB
   ILE MODULE  ILE PROCEDURE
BUPMENUN            1       39         199        390         324          0           0        574          51
BUPINZF             2       39         101        374         322          0           0        556          31
**UNKNOWN           0        0           0          0           0          0           0          0           0
ITMBLDR             2        5           2        170          90          0           0        261          12
CSTBLDR             2        2          13        168          91          0           0        281          12
*INSSDSE          150        7           0        316         181          0           0        247          16
**LICTSKS           0        2         638         11         335          7           1          4          28
**************************************************************************************************************
                             1
ILECLVL2            9        0           0          0           0          0           0          0           0
   ILECLVL2    _CL_PEP
ILECLVL2            9        0           0          0           0          0           0          0           0
   ILECLVL2    ILECLVL2
**************************************************************************************************************
NONILECLVL          9       46         160         17         209          0           0         59         195

**************************************************************************************************************
                             1
ILECLVL4            9        0           0          0           0          0           0          0           0
   ILECLVL4    _CL_PEP
ILECLVL4            9        0           0          0           0          0           0          0           0
   ILECLVL4    ILECLVL4
ILECLVL5            9        0           0          0           0          0           0          0           0
   ILECLVL5    _CL_PEP
ILECLVL5            9        0           0          0           0          0           0          0           0
   ILECLVL5    ILECLVL5
ILECLVL6            9        0           0          0           0          0           0          0           0
   ILEPAYCL    _CL_PEP
ILECLVL6            9        0           0          0           0          0           0          0           0
   ILEPAYCL    ILEPAYCL
ILECLVL6            9        0           0          0           0          0           0          0           0
   ILEPAY      ILEPAY
************************************************************************ ************************************
BUPAP1RN            1        0          67         16           2          0           0         18          13
COMPTIME           68        0           5          0           2          0           0          0           3
*RSLVSP          1440        0          50          0           0          0           0          0           0
*SETACST         1006        0           0          0         116          0           0          0          76
*DEQ              901        0           5          0           0          0           0          0           0
*DBMAINT           19      109          71        171         154          0           0         12           1
*FNDINXEN        1198        0           9          0           0          0           0          0           0
*DESCR             28        0           0         51           0          0           0        100           0
*MODS1            334        0           0          0          22          0           0          0           9
*REQIO            205        0           1          0           0          0           0          0           2
*CRTS              97        0           4          0          24          0           0          0          42
*DESS             118        0           0          0          24          0           0          0          56
*WAITEVT           69        0           3          0           0          0           0          0           0
*SNDPRMSG         248        0           1          0           2          0           0          0           0
*CRTDOBJ           62        0           7          0           0          0           0          0          18
*INSDSEN          185        2           8          4          14          0           0         58           0
*MATPTR          1255        0           0          0           0          0           0          0           0
*LOCK             636        2           0          0           0          0           0          0           0
*RMVINXEN         207        0           3          0           0          0           0          0           0
*MATQAT           722        0           0          0           0          0           0          0           0
*INSINXEN         197        0          11          0           0          0           0          0           0
```

Figure 24. TPST Cumulative Module I/O Breakdown Report

8.2 Benchmarks

Benchmarking can be the most accurate way of obtaining relatively realistic performance data for any type of application. The disadvantage, however, is that it also is the most resource-consuming method of obtaining it. Many times the real environment is oversimplified to overcome this limitation, and the results are meaningless.

Setting up a benchmark requires:

1. The correct AS/400 configuration is available
2. The application and the data are representative and valid
3. That the correct version of all programs and software to be used are available
4. That the required number of users/terminals are available to run the test
5. The transactions are very well defined for each user

Running meaningful benchmarks for interactive workloads is almost impossible without special equipment that allows you to simulate a user in a terminal.

To run a batch benchmark is of course not as complex a task as to test performance of interactive applications, and the first three points above are still valid for this type of test. However, benchmarking concurrent batch and interactive work, which is frequently the actual customer environment, also requires the appropriate number of users/terminals.

Each country has its own procedures for hosting customer benchmarks. In the US, the Rochester lab has a Customer Benchmark Center that offers assistance in benchmarking AS/400 applications. In special cases their benchmark output can be used as input for additional assistance with the SNAP/SHOT service.

For additional assistance from the Rochester Customer Benchmark Center call 507-253-5266 or send a note to RCHVMV(MONTE).

8.3 AS/400 SNAP/SHOT

This section provides an overview of the functions and requirements of the IBM SNAP/SHOT* services available as part of the AS/400 Capacity Planning and Network Design Services. These services include analysis and design of AS/400 systems and networks. Each service is customized to meet specific customer objectives, including performance, cost, manageability, and availability. SNAP/SHOT Services provides the accuracy of a benchmark at a fraction of the cost.

At this time there are no formal SNAP/SHOT offerings outside of the USA. However, in certain countries, the IBM Field System Center may have access to the SNAP/SHOT facilities that would reduce the requirement for on-site attendance in Raleigh.

8.3.1 What Is AS/400 SNAP/SHOT?

- SNAP/SHOT is a discrete simulator tool developed and supported by the Network Systems Services and Support (NSSS) group in Raleigh. Several different system types are supported.
- The SNAP/SHOT tool models the AS/400 hardware and software as a single processor and as a participant in a network of AS/400s.
- It provides the ability to predict capacity requirements and performance for growth and expansion of AS/400 networks.
- AS/400 SNAP/SHOT workshops usually last 3 - 5 days (this does not include preparation time) and are held in Raleigh, North Carolina, USA. (Some workshops may be done remotely by approval.)

8.3.2 When to use SNAP/SHOT AS/400

The SNAP/SHOT AS/400 workshop can be used to accomplish the following capacity planning and network design issues for interactive workloads:

- Analyze the results of downsizing CICS from a mainframe to CICS/400 on the AS/400.
- Rightsizing a CICS/400 application for the AS/400.
- Analyze the results of upgrading to newer AS/400 models
- Analyze the results of downsizing to the AS/400
- Determine the impact of new applications, growth in users, and growth or changes in workload
- Determine optimal network in terms of cost and performance
- Determine the impact of various hardware and communications improvements
- Determine the impact of adding more memory or more DASD on end-user response times
- Load balance current workload across multiple AS/400 Systems to improve interactive response time
- Determine the optimum design of your remote communications sites and lines off of the AS/400
- Determine what size AS/400 system is needed to replace a current S/38 system
- Determine the impact of running 3270 Device Emulation on the AS/400 system in a mixed S/370 and AS/400 environment
- Determine the impact of Distributed Data Management, Remote Job Entry, and Office on the AS/400 System
- Determine the impact of supporting APPC/APPN communications application programs in a network of AS/400 systems
- Analyze results of converting to a Frame Relay Network

8.3.3 AS/400 SNAP/SHOT Customer Workshop Examples

1. Customer Scenario 1:
   - Their main objective was to come up with a ″proposed configuration″ that satisfied their performance criteria and handled growth of users and workload.
   - Many SNAP/SHOT models were run, using various line speeds, SDLC point-to-point versus multipoint, and different AS/400 models.
   - Graphs and charts of the data using GDDM were created to show the results and proposed solutions.
2. Customer Scenario 2:
   - Their main objective was to model their network of AS/400s connected via token-ring using several AS/400 communication functions (APPN and DDM (Distributed Data Management)) and see the impact of increasing workload by 25%, 50%, and 75%.
   - Another objective was to see the impact of changing the user connectivity (remote to twinaxial or token-ring).
   - The customer was able to successfully determine that their configuration will handle up to a 50% growth rate until a major upgrade analysis will have to be done.

8.3.4 How to Sign Up for an AS/400 SNAP/SHOT Workshop

- The system engineer or marketing representative must fill out a SNAP/SHOT nomination on HONE (fastpath of ″NETWORK,″ option HOST SNAP/SHOT).
- Workshop objectives must be clearly defined.
- Conference call with SNAP/SHOT coordinator covering procedures and objectives.
- Complete data gathering requirements and send to SNAP/SHOT coordinator prior to attending the workshop.

For more information US parties can call the SNAP/SHOT information line at (800) 426-4682 (tieline 352-4141). World-wide may send a PROFS note to DALVM41B(SNAPSHOT).

8.4 Performance Tools/400 Agent (5763-PT1, feature H2)

The Performance Tools/400 Agent feature of 5763-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. It is functionally similar to and replaces the V2R3 Performance Tools Subset/400, 5798-RYP.

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

    Create/Delete BEST/1 Model are not supported by the Performance Tools Subset/400, 5798-RYP.

A detailed comparison of the functions of the two products is available in appendixes of the *Performance Tools Guides*.

8.5 Performance Management/400 (PM/400)

Performance Management/400 is both a software product (program 5799-MPG) 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. In the central system with PM/400 installed, that system is then configured to send received PM/400 summary data 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.

In the US there is an ″AS/400 PerformanceEdge*″ enhancement to the IBM AS/400 Maintenance Agreement. PerformanceEdge 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*, SC41-0145-02, for a complete description of PM/400 support.

Figure 25 on page 120 shows an example of PM/400 performance data collection and flow. The example shows the US Competency Center in Rochester, Minnesota, USA as the receiver of the collected data. This is where the data is analyzed and the reports and graphs 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 properly. Contact the country System Product Manager or local IBM representative to determine country-specific support.

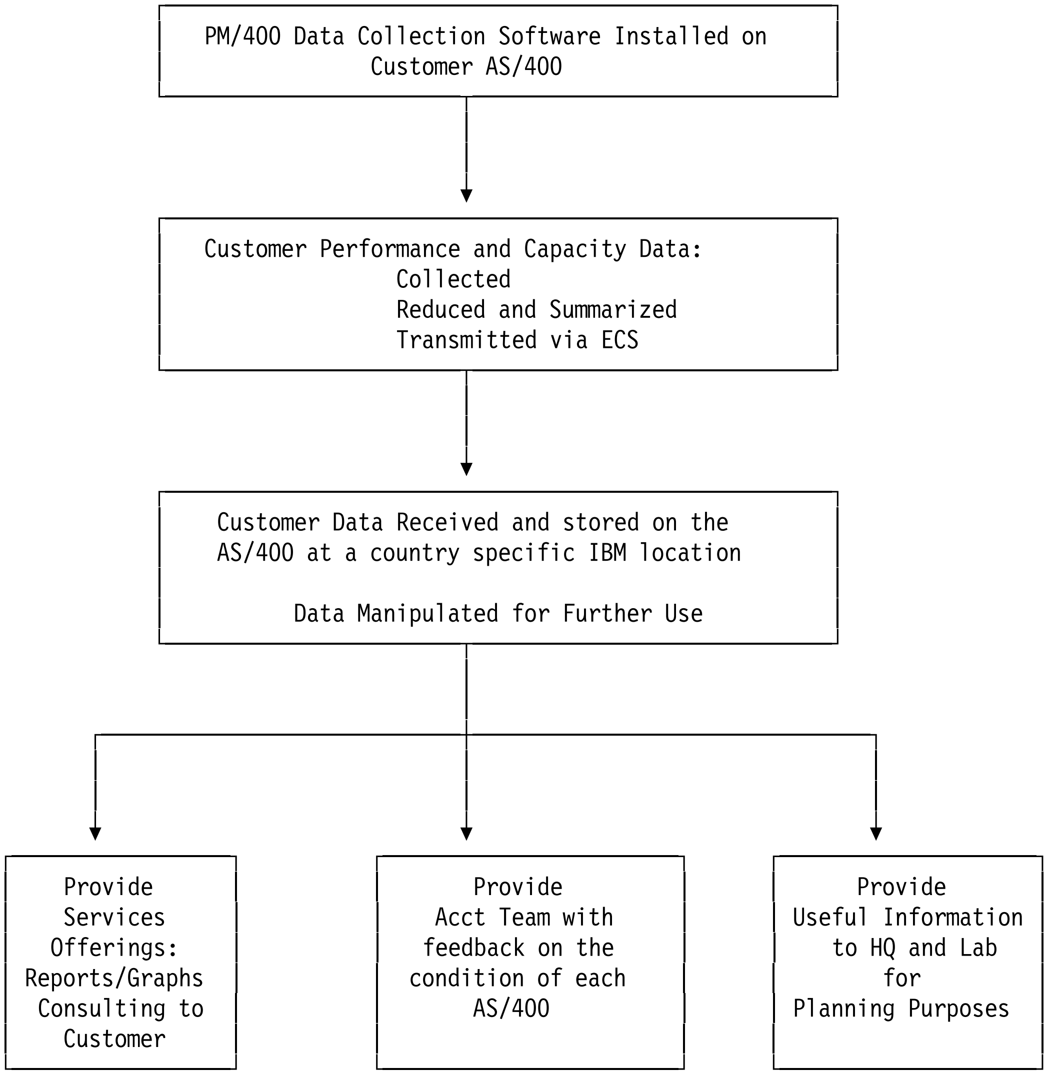


Figure 25. PM/400 Information Flow

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.

8.5.1 PM/400 Levels of Support

If an AS/400 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, DADS) is predicted to exceed the guideline within 6 months. A second letter is sent as a follow up within 2 months of the predicted guideline date.

- PM/400 - Subset Reports and Graphs

  The subset of the total PM/400 service provides black and white (non-color) reports that show resource utilizations per week in an overlay format. This enables the user to determine long term trends

- 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 per day.

  This full level of support 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 kinds of information provided in the reports and graphs is summarized in the following tables.

Table 9 shows available report output for the full service and the subset of PM/400. Table 10 shows graphic output for the full service and subset PM/400 options.

Table 9. 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 10 (Page 1 of 2). PM/400 Monthly Graph Availability

Graph Types	**PM/400 Service Offering: Monthly/Quarterly/Annual (Color)**	**PM/400 - Subset Semi-Annual (Blank/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

Table 10 (Page 2 of 2). PM/400 Monthly Graph Availability		
Graph Types	**PM/400 Service Offering: Monthly/Quarterly/Annual (Color)**	**PM/400 - Subset Semi-Annual (Blank/White)**
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

Reference Figure 26 for an example of the graphic output.

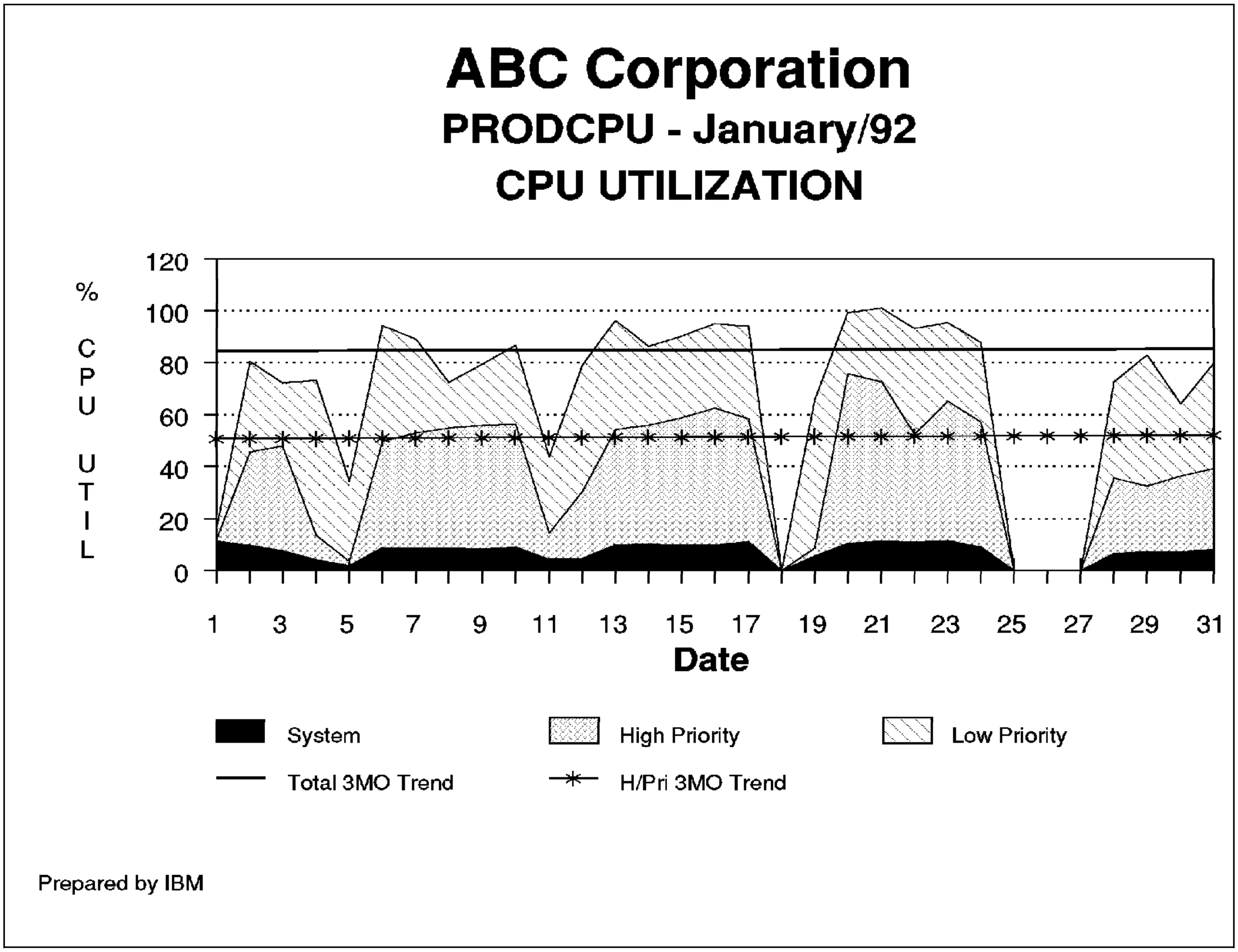


Figure 26. PM/400 Example Graph

8.5.2 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 wish 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 U.S.A. such as AS/400 PerformanceEdge/

8.5.3 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

8.5.4 PM/400 Work with History commands

For those customers with a ″full PM/400 service″ contract, a set of ″work with history″ commands are available for use by those experienced in AS/400 performance analysis. These commands are similar to, but different than, some OS/400 commands like Work with System Status (WRKSYSSTS) and Work with Disk Status (WRKDSKSTS). The ″Work with Historical Active Jobs″ command and associated options provide an easy way to identify user jobs and system work that may be consuming CPU or disk resource above normal expectations.

Figure 27 on page 124 and Figure 28 on page 125 show the last two Work with Historical Job screens in a sequence of PM/400 Work with Historical System Status and Work with Historical Job screens used to detect a job with an abnormally high CPU utilization while that job was active, rather than across a complete Performance Monitor time interval. The sequence is similar to the OS/400 Display Performance Data command screens, but required less key strokes to identify the offending job.

```
                         Work with Historical Active Jobs            XYZ001
                                                                      5/10/94
Interval. . .     16  Library...: PMDEMO      Member...: Q912530755  TUESDAY

End date. .  5/10/94  CPU util..:   100.0  Sys & Int.:  86.9  Act job...:   226
End time. . 11:55:35  Trans.....:     360  Prg *YES..:   398  % Dyn *NO.:  96.2
Elapsed...: 00:14:57  Aux stg...:    5143  % used....:  77.5

Type options, press Enter.
  5=Display detail

  SBS/Job     User       Typ Pl Py CPU% Int  Resp  SynIO AsyIO LglIO PAGf Elaps
  OEPRINT     QSYS       SBS  2  0   .0                0     0     0    0 14:57
   OEPRT001   MAGNATRON  BAT  6 30  3.8              656   138   315    2 14:57
   OEPRT002   MAGNATRON  BAT  6 30  1.8              534   200   424    0 14:57
   OESCHEDLER MAGNATRON  ASJ  6 30   .1               13     1    33    0 14:57
  QBATCH      QSYS       SBS  2  0   .0                1     1     0    0 14:57
   APBCKR     APCLERK    BAT  5 50  1.4              197   159  1311    2 14:57
   ICBEML     DAYOPR     BAT  5 50  1.8               18   509  2835    0 14:57
   ICS0000902 CSPBRH     BAT  5 50   .7              455   107    82    1 14:57
                                                                       More...
F10=Peak system and interactive CPU      F12=Cancel      F13=High priority CPU
F14=Tasks  F15=Work with system status   F16=Resequence  F24=More keys
```

Figure 27. Work with Historical Job - Sort Jobs by CPU while active

To identify the job using the most CPU, place the cursor on the heading **CPU%** and press 'F16=Resequence'.

```
                        Work with Historical Active Jobs            XYZ001
                                                                     5/10/94
Interval. . .     16  Library...: PMDEMO      Member...: Q912530755  TUESDAY

End date. .  5/10/94  CPU util..:   100.0  Sys & Int.:  86.9  Act job...:   226
End time. . 11:55:35  Trans.....:     360  Prg *YES..:   398  % Dyn *NO.:  96.2
Elapsed...: 00:14:57  Aux stg...:    5143  % used....:  77.5

Type options, press Enter.
  5=Display detail

  SBS/Job     User       Typ Pl Py CPU% Int  Resp  SynIO AsyIO LglIO PAGf Elaps
   LONGQRY    LEADER     INT  2 15 24.9              0     0     0    0 14:57
   DSP01      CONSULTANT INT  2 10  9.3  182    .7   901   200  3333   33 14:57
   SLCWS10    SLCNTR2    INT  4 20  6.1   18  51.6   142   105   227   13 14:57
   UTOGDWS1   OGDBRH1    INT  4 20  5.6    7 119.3    17     7    32    4 14:57
   SLCWS8     SLCDP2     INT  4 20  5.6    1 773.0     7     0     7    0 14:57
   WYLARWS1   LARBRH1    INT  4 20  5.5                2     0     2    0 14:57
   SLCWS5     SLCBRH     INT  4 20  5.5    1 769.0    15     0     7    9 14:57
   UTREXWS1   REXBBRH1   INT  4 20  5.5                8     0     3    0 14:57
   COCRGWS1   CRAGBRH1   INT  4 20  5.4                4     0     2    1 14:57
                                                                      More...
F10=Peak system and interactive CPU      F12=Cancel      F13=High priority CPU
F14=Tasks  F15=Work with system status   F16=Resequence  F24=More keys
```

Figure 28. Work with Historical Job - Sequence Jobs by CPU while active

This sequence quickly identified a user running a long running query job (:mini-batch job″) at a priority greater than normal interactive users.

The PM/400 history commands work with any interval of the Performance Monitor and any collection of Performance Monitor data, including the most recently completed interval of an active Performance Monitor.

8.6 Other Performance Services

PM/400 is part of a growing family of support options offered under IBM Performance Management Offerings and Services. Other customer-site options in the U.S. include Performance Examination for the AS/400 system (AS/400 PerformanceEdge), an in-depth analysis of applications, and capacity planning, a detailed model for assessing growth requirements based on current patterns of use.

There are many other IBM services available in the U.S. and around the world that can assist you with virtually every aspect of your information systems environment to help you manage your resources better. For other IBM services contact your IBM representative.

8.7 Performance Investigator/400 PRPQ P84211 (5799-PRG)

AS/400 managers and system operators require tools that can provide immediate and up to the minute information about the resources they manage. For the manager or operator with multiple systems and networks this information becomes even more critical. As systems and networks become larger the data about the critical resources becomes more difficult to gather, analyze and react to. The Performance Investigator/400 combines AS/400 real-time performance data collection with a PS/2-based graphical view of that data. Data is collected from the AS/400 in samples as often as every 15 seconds and then sent to the PS/2 for threshold analysis and display. This tool provides the user with immediate feedback on the performance of their system(s), highlighting problem areas when user-definable thresholds are exceeded, while usually using less than 1% of the host AS/400 resources. Using Windows 3.0, Performance Investigator/400 allows a user to monitor up to 16 AS/400 systems on a single PS/2 display, viewing such data as CPU and DASD utilization, response time and communications bandwidth utilization. Some of the key features of Performance Investigator/400 are:

1. Collection of 9 types of data:
   - CPU utilization
   - Interactive response time
   - Communications IOPs
   - Communications lines
   - Local area networks
   - Disk IOPs
   - Disk arm utilization
   - Machine pool
   - Base/user pools
2. Performance Investigator/400 can be configured to the user′s specifications for threshold monitoring and data refresh.
   - The user can define the thresholds for each type of data or use the defaults provided by IBM.
   - User can define the refresh interval length from as low as 15 seconds.
   - User can define the number of intervals to be displayed concurrently (shows immediate past history) to be between 2 and 20 minutes.
3. Three (3) levels of information are available for each type of data:
   - Level 1 -- This displays several intervals of summarized data (for example, average and maximum) graphically portraying that resource′s performance over time. More detailed data can then be viewed by double clicking on the desired interval and a Level 2 window will appear. Level 1 information for interactive response time and CPU utilization for two different systems is shown in Figure 29 on page 127, where the threshold values are highlighted using a darker color.

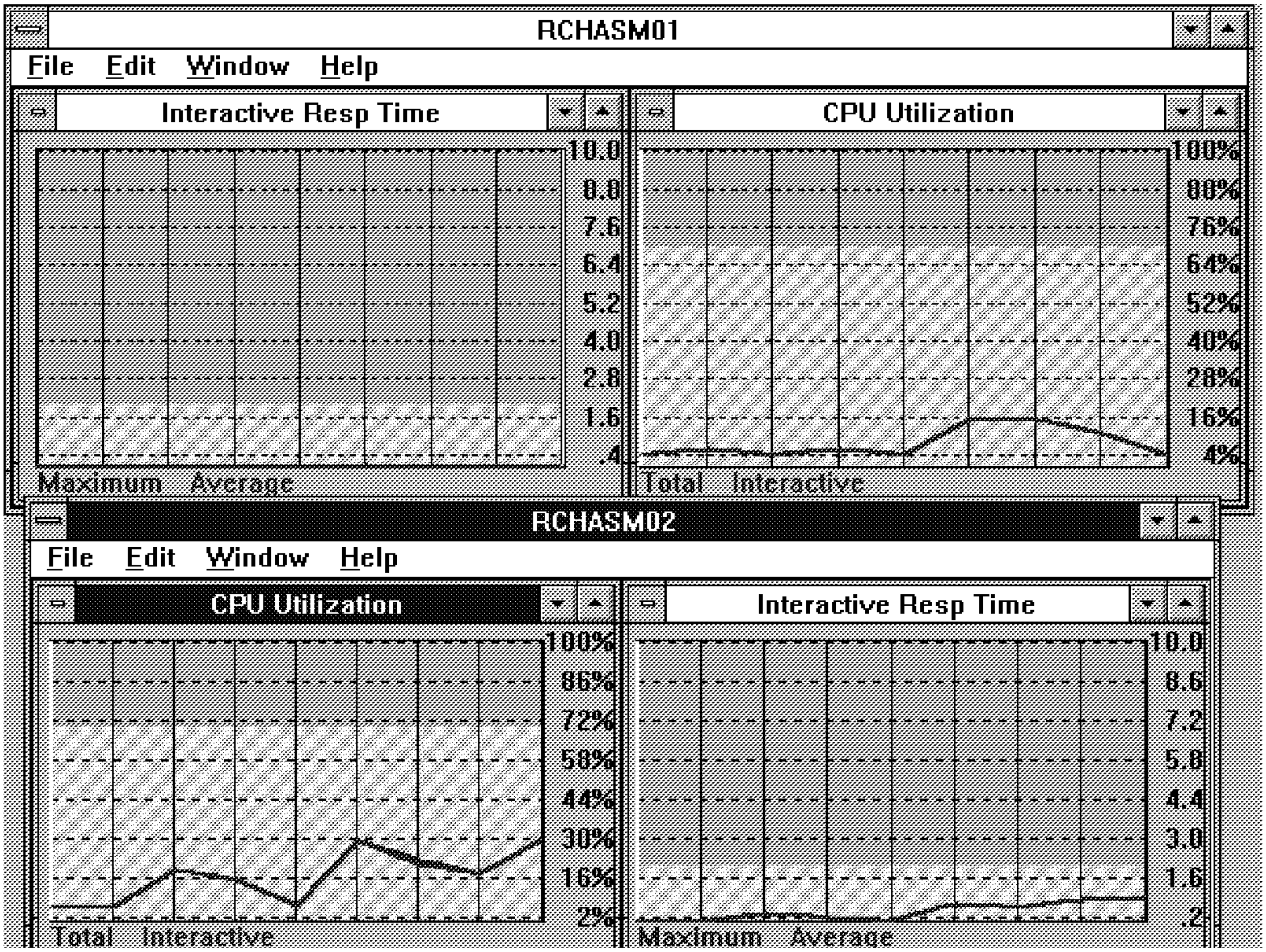


Figure 29. Level 1 Information for Two Different Systems

- Level 2 -- This displays each of the units that were summarized in the Level 1 window. For example, if Level 1 display showed CPU utilization, the Level 2 window would present a bar chart showing the top 10 jobs and their associated CPU usage. To view more detailed information on a particular unit (for example job), double clicking on the desired unit will cause a Level 3 window to appear.
- Level 3 -- This window will display the more detailed information in a list format.

4. All individual windows or views can be reduced to Icons on the screen. The Performance Investigator/400 window itself can also be reduced to an icon. This icon and the icons associated with the Level 1 displays remain active and will change color depending on system threshold conditions.

Pointing with the mouse and clicking in a column of the Interactive Response Time graphic shown in Figure 29 triggers the Level 2 graphic for Interactive Response Time shown in the lower left corner window of Figure 30 on page 128. Double clicking in the first job leads us to the Level 3 information shown in the *Detailed Job Info* window in the same figure.

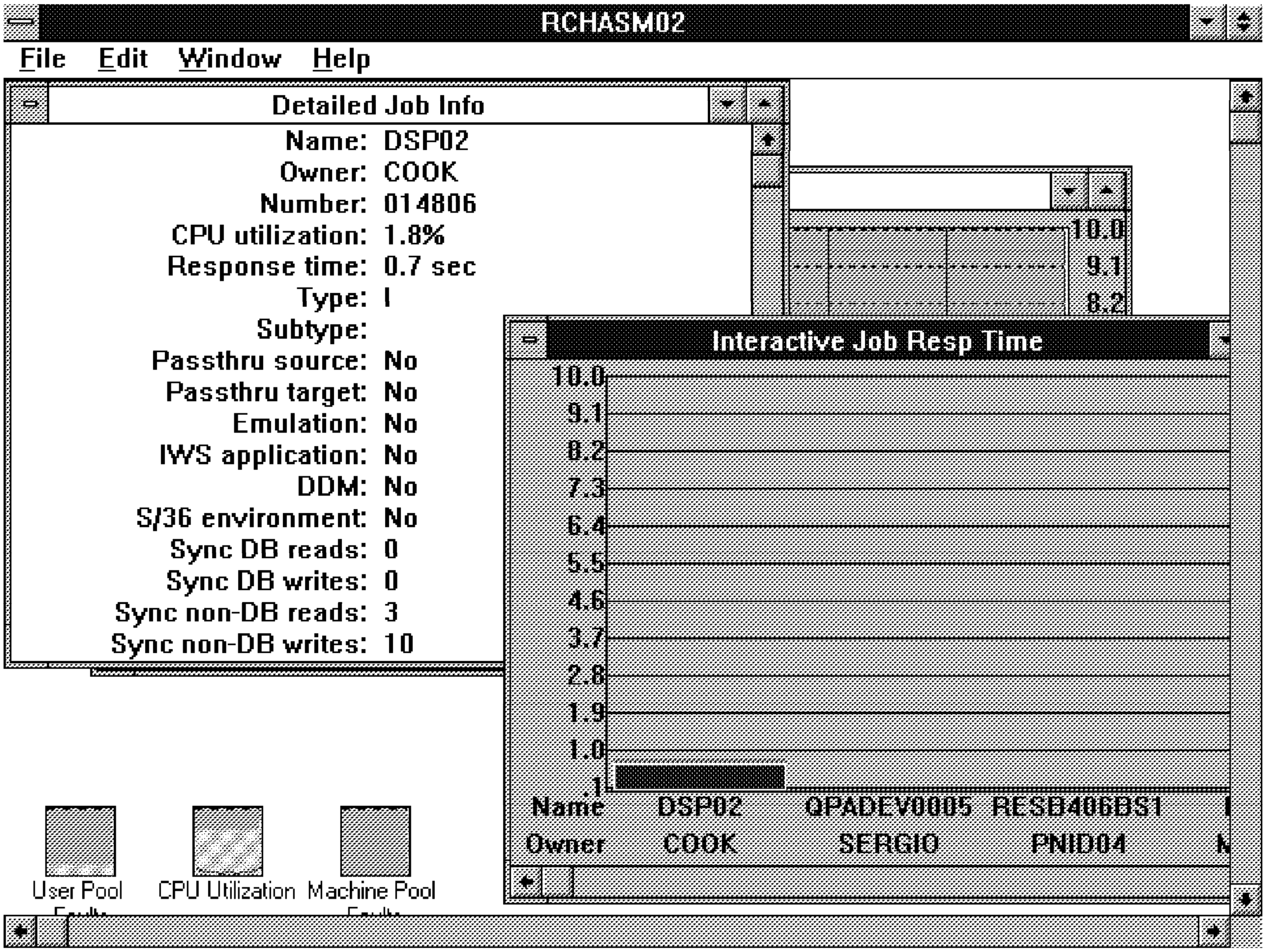


Figure 30. Level 2 and 3 Information for Response Time

Performance Investigator/400 requires functions available with V2R2 of OS/400 and AS/400 PC Support Licensed Program and requires these on the monitored systems as well as Windows 3.0, an 80386SX processor with 2MB memory, and 2MB of available DASD as a minimum on the monitoring PS/2s.

8.8 Queries/Programs

Performance Monitor is part of the OS/400 and stores all the information in externally described database files, which are described in *Appendix A, Performance Data* of the *Work Management Guide*

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 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 very 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.

8.9 Application Performance Tuning Aid (APTA) PRPQ P10110 (5799-XBL)

Application Performance Tuning Aid (APTA) Version 2 is an AS/400 utility. It is used to bind Extended Program Model (EPM) program objects into a single composite (bound) program object. APTA can be used with the EPM compilers C/400 and AS/400 PASCAL to produce program objects with a reduced number of external references. This has the potential of improving the application performance. APTA works through a simple command interface. Debugging capability for the composite object is also provided. APTA is designed for experienced programmers. Users of APTA must exercise caution. An iterative process is often required to achieve the desired results. Even then, improvement in performance is not guaranteed.

For more information see *Languages: Application Performance Tuning Aid Programming RPQ P10093 User's Guide and Reference, SC09-1316*.

APTA is an I-listed PRPQ which means that the licensing requires approval at the laboratory level. For each new user RPQ request submission, your IBM representative must complete a questionnaire.

Recommendation Not to Use APTA

The need for APTA has been obsoleted on releases V2R3 or later. Standard ILE support should be used.

Chapter 9. System Performance Tuning Tips

9.1 Content

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

9.1.1 Areas Covered in This Chapter

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 will be set to default values and the subsystem descriptions (especially the pool definitions) may not be satisfactory for your processing needs.

Note, some subsystem information is also included in Chapter 10, "Design and Coding Tips" on page 189 as the information applies to a particular application, such as ADSM/400.

This section also lists some changes to the list of IBM system jobs in V3R1 as compared to previous releases. The *Work Management Guide - Version 3* 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 7, "Performance Problem Analysis" on page 91.

9.2 V3R1 System Jobs

Unless otherwise noted, there are no V3R1 changes in the functions performed by the system jobs listed.

9.2.1 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.

9.2.2 System Arbiter Job (QSYSARB)

This system job is started by SCPF and provides the environment for running high priority system functions, including:

- subsystem monitor start, end
- tracks ″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.

9.2.3 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 2-10% 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 10.6, “APPC Programming Tips and Techniques” on page 233 for more detailed information in this area.

9.2.4 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.

9.2.5 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 then 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 that 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.

9.2.6 Database System Cross Reference Job (QDBSRVXR)

Beginnning with V3R1 a new system job named QDBSRVXR exists. This job runs program QDBXREF at priority 0 which 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 (eg. RNMOBJ, CRTDUPOBJ). Information about relational database directories (WRKRDBDIRE) and SQL package objects including Distributed Relational Database (DRDB) information is also maintained by this job.

In releases prior to V3R1, these functions were performed under the QSYSARB system job.

During the first IPL after V3R1 installation job QDBSRVXR runs to convert the previous release implementation database cross reference files to new V3R1 capabilities under DB2/400. On a system with a very large number of files and fields (and SQL tables and columns) this conversion of cross-reference information from a pre V3R1 release to the first Version 3 (V3R1 or later) installation could take very long and not finish all conversion work until subsequent IPLs. See “V3R1 One Time Cross Reference File Conversion” on page 213 and 9.2.7, “Database Server Jobs (QDBSRV01... QDBSRVnn)” for additional performance considerations for job QDBSRVXR.

9.2.7 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 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 releases previous to V3R1, there was a maximum of one of these jobs active. In V3R1 there is one of these jobs per processor.

In V3R1, 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 209.

Note that as part of the conversion of database reference information to Version 3 format, a QDBSRVnn job will identify files/table data to be converted by the QDBSRVXR job. Soon after the install of the new release completes, one of the low priority QDBSRVnn system jobs will run 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 will reflect the current information from all of the database files on the system. All database files that exist in a library which existed during the install will be eventually be processed by this new cross reference conversion function.

This conversion may continue across IPLs until it completes processing all of the libraries, and it will wait forever to obtain any lock before continuing with the next database file or library. Because of the locking implementation, applications which 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 “V3R1 One Time Cross Reference File Conversion” on page 213 for additional information.

The cross reference conversion process also has system disk storage size considerations. The full description of cross reference information conversion is contained in an ″information only″ APAR II08311.

9.2.8 Decompress System Object Jobs (QDCPOBJ1... QDCPOBJn)

These jobs decompress or compress system objects as needed and are started as part of IPL.

No changes to these jobs for V3R1.

9.2.9 System Job Scheduler Job (QJOBSCD)

This system job performs the OS/400 job scheduling functions as specified by the Add Job Scheduler Entry (ADDJOBSCDE) and Change Job Scheduler Entry (CHGJOBSCDE) commands.

QJOBSCD is started by the system as part of IPL and monitors time and date and scheduled job parameters based on the job scheduler entries.

Note that V3R1 SystemView System Manager/400, via its change management support also provides job (change request) scheduling with additional capabilities to make the start of a subsequent change request dependent upon the results of the previous request.

SystemView System Manager functions work in IBM supplied subsystem QSYSWRK. Full performance management of SystemView System Manager/400 is beyond the scope of this redbook.

9.2.10 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 will show various ″status values″ based upon the function it is performing - deleting no longer need 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 spool cleanup as part of IPL after an abnormal system termination.

9.2.11 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 IPL and remains active until system shutdown.

9.2.12 System LU6.2 Resynchronization Job (QLUR)

This system job is new for V3R1 and performs any *two-phase commit* processing.

QLUR is started during IPL and remains actice until the system is shutdown.

9.2.13 System Integrated File System Jobs (QFILESYS1... QFILESYS2)

The Integrated File System (IFS) is introduced with V3R1. One or two system jobs QFILESYS1-QFILESYS2 ensure changed data accuracy and file cleanup for the IFS support. This includes ″pre-bringing″ data into main storage and forcing changed data to disk.

QFILESYS1 is started during IPL and remains active until the system is shutdown.

Note that VLIC tasks BAIT01-BAITnn actually perform the work and can be seen in the WRKSYSACT screen output and the Performance Tools/400 Component report.

9.3 V3R1 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 the *Work Management Guide* for detail information.

Many of the IBM-supplied subsystem monitors have routing entries pre-defined 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 *Work Management Guide - Version 3*, GG24-3306.

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.

9.3.1 QBASE subsystem

This subsystem can be the controlling subsystem on the AS/400 (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, etc. 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 start up program that is called soon after the controlling subsystem has started at the end of IPL.The startup program starts the subsystem monitors both IBM and the customer wish to have active after IPL. As shipped from IBM, the startup program starts subsystem monitors QBATCH, QINTER, QSPL, QSNADS, and QCMN. In V3R1 the shipped program also starts QSERVER (used by Client Access/400.

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.

9.3.2 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 which 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 Work Station Function (WSF) sessions, RUMBA/400 sessions, and OS/2 Communication Manager 5250 emulation sessions, and ASCII Workstation Controller display devices.

9.3.3 QCMN subsystem

QCMN is shipped to ease set up for receiving remote program start requests that start up communication-based jobs on the AS/400.

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, QCMN subsystem contains routing entries for Client Access/400 jobs. See 10.13, “Client Access/400 Performance Tips” on page 286 for more information on Client Access/400 use of subsystem QCMN.

See also, 10.15, “ADSTAR Distributed Storage Manager/400 Performance Tips” on page 308 for information on ADSTAR Distributed Storage Manager/400 (ADSM/400) use of subsystem QCMN.

9.3.4 QSPL subsystem

This subsystem is shipped to control all spooled output work, such as the ″Start Printer Writer (STRPRTWTR) command jobs.

9.3.5 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

  In V2R3 this job provides all SNADS routing services for sending and receiving distributions with remote systems.

  In V3R1 this job continues to provide SNADS routing for *change management* functions. However, routing functions for normal SNADS distributions (for example documents and ODF) are moved to the QMSF job(s) 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 then you will also have a corresponding QGATE/remote-location-job active when System Manager/400 or Managed System Services/400 active.

- QGATE/TCPIPLOC

  This job is activated when TCP/IP Simple Mail Transfer Protocol (SMTP) is activated for the local system.

9.3.6 QSYSWRK Subsystem

The QSYSWRK subsystem was introduced with V2R3 to be a common subsystem for various system jobs. In V3R1 additional jobs are placed in this subsystem. 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 only **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 V3R1 ″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 *ManageWare/400 Administrator′s Guide*, SC34-4478.

- 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 GG24-4378 and this redbook in 10.14, “LAN Server/400 Performance Tips” on page 298

- 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 - Version 3*, SC41-3321. For information on SystemView Managed System Services/400 refer to *SystemVIew Managed System Services/400 Use - Version 3*, SC41-3323.

  – 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 V3R1 SystemView System Manager/400, V1R5 or later NetView Distribution Manager, etc.)

    - 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 could be multiples of these jobs. You may 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 V3R1 SystemView System Manager/400 if Start Manager Services (STRMGRSRV) command has been issued on this local system.

– QSVSM/QNSCRMON

  This job monitors the change management requests and initiates the sending of these requests to the managed system. This job should be active only if V3R1 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 of mail or SNADS distributions

- TCP/IP support

  In V3R1 TCP/IP support is included in OS/400. When the command Start TCP/IP (STRTCP) command is issued, several jobs are started in QSYSWRK. For previous releases, TCP/IP work was performed in subsystem QTCP. This subsystem no longer exists in V3R1.

  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 V3R1 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 *IMMEDiate, all jobs will abnormally terminate and cause some system overhead in generating job logs. Re-issuing STRSBS QSYSWRK will automatically restart all the jobs discussed previously except the TCP/IP jobs. You must do ENDTCP and follow with STRTCP to make TCP/IP support operational again

9.3.7 QLPINSTALL

This subsystem performs Licensed Program (LP) installation functions.

9.3.8 QPGMR

This subsystem is available for application development functions.

9.3.9 QSERVER

This subsystem is shipped with V3R1 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 below:

- user-id/QPWFSERV

  File serving support includes storing programs and files as a network drive (″virtual disk″) for the attached client.

- QPGMR/QSERVER

  Thi auto start job sets up the file serving and database serving environment on the AS/400

- QUSER/QZDAINIT

  There is one of these database serving (remote ODBC and SQL) functions for each active client session. QZDAINIT is implemented as a prestarted job.

For more information on these server jobs refer to 10.13.2, “Client Access/400 Work Management” on page 292.

9.3.10 QADSM

This subsystem is active when ADSTAR Distributed Storage Manager/400 (ADSM/400) is installed and active. Most ADSM functions run and 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 10.15, “ADSTAR Distributed Storage Manager/400 Performance Tips” on page 308.

9.3.11 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 redbook *Backup Recovery and Media Services/400 Implementation Tips and Techniques*, GG24-4300.

9.3.12 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 10.16, "LANRES/400 Performance Tips" on page 314.

9.3.13 QAUTOMON

QAUTOMON subsystem is provided with SystemView OMEGAMON Services/400, program number 5738-ES1. QAUTOMON runs the OMEGAMON ″condition″ monitoring jobs and the jobs for the AUTOMATED FACILITIES/400 feature of 5738-ES1.

The SystemView OMEGAMON Sevices/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/400s 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 redbook *Managing Operations on AS/400s with IBM SAA SystemView OMEGAMON Services/400*, GG24-4136. Some performance tips are included in this redbook under 10.7, "SystemView OMEGAMON Services/400 Tips and Techniques" on page 236.

The following is 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.

9.4 Review System Values

There are some system values that affect performance, such as QMCHPOOL, QPFRADJ, QACTJOB, QTOTJOB, QDEVRCYACN, QCMNRCYLMT. Review these values because they could be related 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 QMCHPOOL size according to the *Work Management Guide* 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 will notice a high fault rate in pool 1 and this severely impacts all jobs throughout the system.

  See 3.6.5, "Working Set Size" on page 26 for specific V3R1 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, then 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 will allocate 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, will also cause overhead as space for 100 new jobs will be created as soon as there is one active job over the QACTJOB value.

  Note that you need to keep QACTJOB, QTOTJOB, QADLACTJ, and QADLTOTJ at reasonable values. If you make QACTJOB and QTOTJOB excessively high, 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 could be 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 provided the following cautions are followed.

  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.

  If you have a high number of spool files when using WRKSYSSTS and add 20% more to set the QTOTJOB value, this will significantly add to IPL time. Performance will also degrade 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 will be active only after the next IPL.

- QRCLSPLSTG: When a spool file is printed or deleted from a spool output queue or an entire spool 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 spool file storage allocated″.

  *NONE may also be specified for QRCLSPLSTG. **However, *NONE should not be used.** *NONE means space is de-allocated immediately. *NONE can cause poor performance since the system will never be able to re-use existing space and will have to create a new database member and allocate storage every time a new spooled file is opened by the system. This create and allocate space overhead slows down the affected job.

  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 2 additional sets of communication lines or control unit retries within a 5-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 would be made only for a specific line or control unit.

Incorrect setting of a QCMNRCYLMT/CMNRCYLMT value could cause the system to perform this ″second level″ of 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 could consume significant system resources. If this occurs, stop the process by varying the object off.

- QDEVRCYACN: The default for this value is *MSG, which will result 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 will abnormally end the job with a 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/job has the same error detection considerations for QDEVRCYACN(*MSG).
- Use system value QDSCJOBITV for controlling how long disconnected jobs will remain in disconnected status. When the QDSCJOBITV time limit is reached, the system will automatically end 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 will have subsystems QBATCH, QCMN, QCTL, QINTER, QSNADS and QSPL started automatically.

9.5 Consider Dividing the Main Storage into Separate Pools

9.5.1 The Reason for Separate Pools

Most AS/400 environments run interactive, batch, and spool 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. Use of 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.

9.5.2 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 will check the QINTER subsystem description.

Enter command ″DSPSBSD QINTER″ 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 will have to create another one by using command ″CHGSBSD SBSD(QINTER) POOLS((2 *INTERACT)″.

Pool number 1 is the pool in which the subsystem monitor will run. To verify that all the jobs in QINTER go to pool number 2, choose option 7 to display routing entries. If they point to pool number 1 enter command ″CHGRTGE SBSD(QINTER) SEQNBR(nnnn) POOLID(2)″ for each routing entry. Replace ″nnnn″ with actual sequence numbers. In releases prior to V3R1, this command may run only when the subsystem is inactive. Beginning with V3R1 CHGRTGE may be issued while the subsystem is active and affects jobs that start after the change has completed. After this is done enter command WRKSHRPOOL 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 6 jobs doing work in the memory pool at any given time and all these jobs have 1666KB of memory to use. Please refer to Table 29 on page 325 and to the *Work Management Guide* for the 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 are 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 in which all the printer output is produced. Enter command ″DSPSBSD QSPL″ 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 you enter the ″WRKSHRPOOL″ command to verify that pool *SPOOL has memory size and activity level. A reasonable amount would be 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 storge pool 4000KB (4MB). For more information about the size and activity level settings of *SPOOL refer to the *Work Management Guide*, SC41-8078. Please refer to the Table 69 on page 348 and to Table 31 on page 325 for the 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 use of the SETOBJACC command and expert cache support. See Appendix E, "OS/400 Expert Cache and Set Object Access Overview" on page 441 for additional information.

9.6 Additional Considerations

9.6.1 Separate Subsystems

The following are two examples of 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 would be to have local twinaxial or ASCII workstation jobs in one subsystem and all other interactive jobs in another subsystem. This second set of jobs could be those on LAN lines or remote communication lines.

The separate interactive subsystems could 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 would not affect the throughput of the local workstation users. You should also consider placing batch jobs to a separate pool in the daytime because batch jobs normally take more system resources than interactive jobs.

9.6.2 Separating QBATCH from *BASE

An important factor of improving the performance is to separate batch jobs from all the other jobs running on the AS/400.

To estimate the amount of memory needed by a job, bear in mind that one query requires about 500KB of memory and one compile requires about 1000KB of memory. For example, if you intend to compile programs and run queries during daytime the initial amount of storage for *SHRPOOL1 would be 1500KB and activity level would be 2. Please refer to Table 33 on page 326 for the accurate values to use.

If most of your batch jobs are run at night you might want to make sure that they will be ready next morning when people come to work. The trick to use is that in the evening you enter ″CHGSBSD SBSD(QBATCH) POOLS((1 *BASE)″ and after that you terminate all subsystems not needed during the night. Then QBATCH will have more memory and the job throughput should be better.

9.6.3 Batch-like Jobs

There are many jobs that should be run in a batch mode because of their considerably heavy use of system resources. Good examples are save/restore operations, cleanup jobs(CLROUTQ), query jobs, etc.

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.
- 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 you just enter command SBMJOB 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 screen and press Enter again to submit the job. You can monitor the status of your jobs with command WRKSBMJOB or use command WRKUSRJOB to monitor all the batch jobs in the system no matter who submitted them. Please do not get confused when job type CMNEVK appears on the screen because in fact it is a batch job started by Work Station Function of Client Access/400

9.6.4 Communications Subsystem

You should consider creating an additional memory pool for the communications subsystem QCMN if you have many PC Support jobs or if you use a lot of 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 which tend to use a considerably large amount of system resources. One way to find information about timeouts is to start service tool (STRSST command) and search for communications errors in the error log. If there are messages that are not self-explanatory please contact IBM service for help.

9.7 How to Use System Provided Tuning

The following 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.

Note that each new release of OS/400 may provide additional automatic tuning of the system.

9.7.1 Automatic Tuning: QPFRADJ

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 the pool from another shared or system pool that has acceptable page faulting values. If there are several pools with high fault rates please 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 operations. Review the *Work Management Guide* for more details on this 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 operational environment, turn off QPFRADJ automatic tuning by setting QPFRADJ to 0. If there are mysterious slowdowns with the response times please 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 the *Work Management Guide*. For example 15000KB of memory and activity level 9 would be a good starting point for an AS/400 with 48MB of memory. Please refer to Table 30 on page 325 for the accurate values to use.

For V2R3 PC Support/400 shared folder Type 2 support, automatic tuning **may** cause the machine pool to be set too low

For V3R1 Client Access/400 *original client* support capability will continue to use shared folder support. Consider using automatic tuning when these functions are not active. Then set QPFRADJ to 0 and manually monitor and set the machine pool size for the ″typical Client Access/400 or PC Support/400 activity″ environment.

For additional PC Support and PC Work Station Function performance information, refer to *PC Support/400 Implementation and Performance*, GG24-3636.

For additional Client Access/400 performance information, refer to 10.11, "Client Server Performance Tips and Considerations" on page 276 and the *V3R1 Performance Capabilities Reference*, manual, ZC41-8166 February 1995 or later.

Note that V3R1 QPFRADJ includes logic changes in the machine pool algorithm to more precisely manage maximum pool size and to more quickly free up unneeded machine pool storage. Equivalent function PTFs were made for earlier releases during the last quarter of 1994:

- V2R3: SF17949
- V3R0.5: SF17954

The V3R1 QPFRADJ job also uses the new page fault guidelines discussed in 3.6.6, "Memory Guidelines" on page 28 This support is included in PTFs for earlier releases:

- V2R2: SF18188 and co-reqs
- V2R3: SF18424 and co-reqs
- V30M5: SF18189 and co-reqs

9.7.2 Using QTSEPOOL

See 9.4, "Review System Values" on page 143 for details.

9.7.3 Performance Tools Advisor

The Advisor output is grouped under the following headings: Recommendations, Conclusions and Interval Conclusions.

The Advisor will make 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 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 would not make a change would be 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 could be an example of when a recommendation would not be 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 ″Advisor″ chapter of the *Performance Tools Guide* for detailed information.

9.8 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 by *IBM Performance Tools/400* . Also, refresh the display at about 2 minute intervals only, because 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 LIC tasks. If the NDB page fault rate in the machine pool (pool 1 or *MACHINE) is greater than or 2 to 5 faults/sec on a system performance may be improved by increasing the size of the machine pool, thereby reducing the page fault rate. Please 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 *AS/400 Work Management Guide*, *AS/400 Performance Tools Guide* or Appendix A, "Guidelines for Interpreting Performance Data" on page 323 for more information on paging rates. Also refer to index entries in this redbook for *memory* and *storage pools* for more information.

LIC tasks that support workstation I/O run in the machine pool and can become a bottleneck if the pool′s fault rate is too high. Refer to the *AS/400 Work Management Guide* or *AS/400 Performance Tools Guide* for guidelines on setting machine pool size. For example, faulty workstation cabling can increase the page fault rate. If there is lots of faulting in the *MACHINE pool please start the System Service Tool and check if there are several error log entries for local work stations. If adding memory to the machine pool doesn′t reduce the page fault rate, do not continue to increase the size of the pool.

On systems that are performing well, the page fault rate in each storage pool is usually less than 10 to 20 NDB faults/sec. 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.

Concurrent Save/Restore in either the batch or interactive subsystem will probably increase paging because the storage it uses in the pool is not available to other jobs. If Save/Restore runs at a high priority, it can cause a noticeable impact on overall system performance for a period of time.

Portions of the **Create Logical File** also make some pages unavailable for a period of time, though not to the extent that Save/Restore does.

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. How would you do this? In the case of **batch jobs or non-interactive jobs** you could run each job in its own storage pool. Since storage management steals pages only from the job′s pool and since the job is the only one in the pool, it cannot get another job′s pages. If the job is stealing pages from itself that it needs again in a short time (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 a reasonable 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 Work Station Function and powering off the PC with several WSF sessions active. In large configurations this could involve over 100 jobs abnormally terminating at the same time.

The system attempts to control this situation by lowering the priority of the job termination processing, but it is still a significant impact to the overall system performance.

For incoming program start requests (target job initiation) from communications devices, consider using prestarted jobs (Add Prestart Job Entry (ADDPJE) command). Properly designed prestarted jobs have already completed job initiation before the request is received and do not actually terminate when the conversation ends. Therefore, requests to use a prestarted job will 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.

9.9 Activity Level Performance Considerations

In analyzing activity level performance considerations you need to understand that:

- You should use the recommended settings in *AS/400 Performance Tools Guide* or Appendix A, “Guidelines for Interpreting Performance Data” on page 323.
- Changing the activity level value will 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 the 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. See *AS/400 Performance Tools Guide* or Appendix A, “Guidelines for Interpreting Performance Data” on page 323 for information on setting activity level values.

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 could result in an unrealistic activity level value. For example, with a B70 interactive pool, 60MB/300KB would give an activity level of 200. If your B70 response time was two seconds, this activity level setting would allow you to run up to 100 transactions per second or 360,000 transactions per hour!

Reasonably speaking, many machines in this class might get 30,000 to 40,000 transactions per hour or 8-12 per second average and 20-25 per second peak load. The effect of setting it much higher than the normal peak load allows almost every transaction arriving during a peak load situation to start processing as soon 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** will result 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, etc.). The real constraint to activity level settings are high pool paging, repaging (″thrashing″), and the resulting disk I/O usage. Once you find the proper value, set it and forget it.

9.10 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 you only make changes that affect the resource (disk, CPU, long waits, etc.) you are dealing with.

Important: Make only one change at a time, and then continue with the measurement, following the cycle.

If your problem has been solved, then keep the performance data for future requirements.

If problems remain after making a change, you need to go deeper into problem analysis and will probably need to collect performance monitor trace data (STRPFRMON TRACE(*ALL)).

9.11 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 *Work Management Guide* 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 323.

Note: You have to repeat steps 4 through 7 for all the other pools in your AS/400; step 3 is for *MACHINE pool only. Follow the roadmap during periods of high systems activity because it is of no use to tune the system when there are no jobs in the system.

```
              Interactive AS/400 Tuning Roadmap

 1. Enter command WRKSYSSTS

 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) -50K if fault rate = 0
       2) +50K if fault rate > 3.0
       3) Press PF10 to reset and go to step 2

 4. Is the DB fault +NDB fault > 20 in any pool?
    a. Yes ... Increase pool size by 50KB, 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 10x 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.
```

Figure 31. AS/400 Tuning Roadmap

9.12 Operating Environment Tips

IPL your System: Occasional IPLs are no longer required on the AS/400 for optimum operation. However, it is at IPL time that permanent and temporary addresses are regenerated, permanent and temporary job structures are built and so on. Refer to *AS/400 Operator's Guide* for more information but try to IPL at least once a month.

Note that temporary job structures are built during run time as needed, according to system values QADLACTJ and QADLTOTJ described earlier in 9.4, "Review System Values" on page 143.

Storage Pool Tuning: With Version 2, let the system do automatic tuning of shared pools by setting system value QPFRADJ to either 2 or 3 (value 3 is usable from Version 2 Release 2 onwards. 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 use of 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 *Work Management Guide*, *Performance Tools Guide*, or Appendix A, “Guidelines for Interpreting Performance Data” on page 323 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 greater than 15 per second 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 Work Station 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 via the option 2 from Start Service Tools (STRSST) or the Printer Error Log (PRTERRLOG) command. Some errors will be self-explanatory and others will require IBM service assistance.
- Work with System Activity (WRKSYSACT) command. If screen 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, and/or high disk IOP utilization, and/or 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 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 expert cache support to have the system identify objects that should be loaded and held in the specific shared storage pool. These facilities are offered for the larger main memory models of the AS/400, such as the D70 and D80. 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 main storage resident and the repeated use of the command can cause a set of objects to be resident in a main storage pool. For more information about using this command please refer to *Control Language Reference* and *Work Management Guide* manuals and Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 441

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 will ″pin″ the object in storage during this heavy usage, thus eliminating 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.

For more information, refer to Appendix E, “OS/400 Expert Cache and Set Object Access Overview” on page 441.

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.

System commands Work with Active Jobs (WRKACTJOB) and Work with System Activity (WRKSYSACT) 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 via the ENDJOB command. If ENDJOB ... *IMMED does not complete, then use the End Job Abnormal (ENDJOBABN) to terminate the job. **Caution, ENDJOBABN will cause 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, etc. Corresponding messages are sent to message QSYSMSG, if it has been created by the user. Also, in V3R1, new ″Service Attributes″ (use DSPSRVA/CHGSRVA command) include the user profile class (multiple users) that will receive 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 message include the following and 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 will stop 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, for V3R1 whether ILE capabilities are being used. Optimization options enable some form of reduction in the number of IMPI 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.

In the non-ILE environment the program is compiled so that register-based operations are used wherever possible. With ILE, additional optimization can be achieved.

During ILE program development, we recommend using OPTIMIZE(*BASIC) on the create command as this has minimal impact on compile time and has some improvement on run time.

An ILE CRTxxxMOD command provides three OPTIMIZE options - *NONE, *BASIC, and *FULL.

Optimization has a wide range of possible performance effect. Compute intensive programs with significant looping, arithmetic operations, and string handling may experience 2 or more times improvement. Programs that use little of this kind or processing but perform significant I/O operations may receive little benefit of optimization.

Note that running the Sampled Address Monitor commands against an optimized program will not accurately identify HLL program instructions causing high system resource consumption.

Reserve the maximum optimization option for those compute intensive programs or large, high usage programs. Have a record of performance before optimizing as programs that do excessive data manipulation may, in some cases, result in worse performance when optimized.

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), etc. 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 will 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 would be to run the function interactively but lower the priority of the current job (CHGJOB command) before running this function.

Note that 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.

The Transaction Report - Job Summary printed with option *HV will show LIC tasks. There will be an LDxxxx task ID shown for each Save/Restore action. Analysis of the PRTTNSRPT the QTRJSUM file can indicate how frequently Save/Restore was performed while the Performance Monitor was running.

Submit Queries to Batch: Query/400 supports submitting the defined query as a batch job. Enter the SBMJOB command and press PF4 to enter command RUNQRY with all the parameters. After that press Enter twice to submit your query. This is recommended for long running queries to reduce their impact on interactive resources.

9337 and 6530/6502 IOP Disk Performance: Several models of the Version 2 systems and the Advanced System Series models support the 6501 disk I/O Processor (IOP) which can attach the faster disks of the 9337-2xx and 9337-4xx series disk models. The Advanced System Series also supports the 6530 and 6502 disk IOPS which support fast disk (6602, 6603, 6607, and 6608) attachment.

These disks and disk IOPs are recommended for best price performance. When attached to the 6501 and 6502 disk IOPs the newer disks can deliver significantly improved performance over the older 9332, 9335, 9336, and non 2xx and 4xx 9337 disks.

Even with RAID-5 enabled, the new disks deliver significant performance performance improvements over the 9332, 9335, 9336, and non 2xx and 4xx 9337 disks.

The key concern with the newer disks is that in order to minimize cost of installation the customer may select a disk with a large number MB of storage which reduces the number of disk arms. Since in general, the more arms the less likely a heavy disk I/O environment will reach a bad performance threshold of I/Os per second per arm.

Refer to 3.7.10, "Disk Guidelines" on page 36 for more information.

Use Mirroring: When mirroring is active, the system looks at each read request and chooses the unit which is least busy and/or would have the shortest seek. Writes are executed twice - once to each arm. In a general case mirroring will add a small amount of system overhead. In a primarily "write-oriented" environment, the overhead of mirroring can become significant. 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 significantly lower cost protections and deliver excellent performance it the appropriate disk IOPs and newer disk devices are selected.

Checksum: Checksum protection is a function that protects data stored in an auxiliary storage pool from being lost because of a single disk failure. When checksum protection is in effect and a disk failure occurs, the system reconstructs the data during IPL following replacement of the disk unit.

With checksum protection each permanent write involves three additional disk I/Os and some additional processing. The steps involved in a permanent write operation with checksum protection active are:

- Read the original page of data from disk.
- Determine changes made to the page.
- Write the updated page to the disk.
- Read the checksum image from disk.
- Create the updated checksum image.
- Write the updated checksum image to disk.

With the availability of 9337-2xx, 9337-4xx disk and associated disk controller and the Advanced System Series 6502 I/O Processor attached disk capabilities to provide RAID-5 disk data protection at very reasonable price/performance, system checksum protection is not recommended.

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/job user profile to determine 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 Table 34 on page 326.

As the number of authority lookups increases, CPU utilization can increase significantly. Beginning with Version 1 Release 3 a change was made to reduce system overhead for users with extensive private authorities. Objects created prior to V1R3 that are now on a V1R3 or later system, will be updated to the new support when a private authority is granted to or invoked from an object. If no private authorities on the object are changed, the object will not use this new support.

Up through Version 2 Release 2, a program QSYFIXAUT was provided to change object authority information to use the new support made available with V1R3. QSYFIXAUT is no longer available starting with V2R3 and there is no replacement function.

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,″ the processing of the user profile can become excessive when the private authority is less than public authority.

- Use of 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 could increase as discussed above 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.

  Note that with Version 3 Release 1 new **supplemental groups authority** and **primary group authority** require additional authority lookup performance consideration.

  Supplemental groups enable a user to be part of more than one group profile. This is supported via 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 profiies.

  Both the *AS/400 Security Reference - Version 3*, SC41-3302, 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.

Use of the Integrated File System (IFS) for V3R1: Use of IFS interfaces is not completely implemented for all V3R1 OS/400 functions and licensed programs. Although the use of IFS offers the potential for improved performance, improved performance should not be assumed until test results for specific functions are documented.

For this redbook edition all performance information related to IFS support is included within 10.11, “Client Server Performance Tips and Considerations” on page 276.

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, 2 seconds is acceptable, because interactive transactions *should not* last longer than 2 seconds. However, there are cases during heavy interactive workload, where lowering the time slice to .5 seconds may improve performance.

Using *Perfomance Tools/400* helps you to decide the value of TIMESLICE. From the System report you get average CPU seconds per interactive job type; multiply the number by three and set the result to TIMESLICE value.

Consider setting interactive job TIMESLICE to .5 or less seconds on the faster AS/400 processors, such as the B60 and up.

Use DEVRCYACN or Program for Work Station 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 will detect this looping after a period of time and automatically terminate the job at a reduced priority. However, programming for workstation errors and or use of 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 screen. 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 screen 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 then loop with failing workstation I/O operations if the error message or major/minor return code is ignored. However, data management will automatically detect five successive I/O failures and automatically terminate 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.

System value QDEVRCYACN can be set to a value that will be 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 that in a completely uncontrolled environment, the suspended jobs resulting from *DSCMSG may keep files open and retain record locks which may prohibit other jobs from running. System value QDSCJOBITV can be used to end the disconnected jobs after a period of time. The WRKCFGSTS and WRKSYSACT commands will not show disconnected jobs. The WRKACTJOB command will show the disconnected jobs if function key 14 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.

Note that in some environments Display Station Pass-Through is used to run several ″menu options″ on a remote system with the intention of leaving the local workstation operator unaware that pass-through is being used. Prior to Version 2, application programming behind the menu could make the initiation of pass-through transparent to the local operator, but returning control to the local program required the operator to use System Request menu support to enable the return.

Use of the Transfer Pass-Through (TFRPASTHR) command within a program can make return to the local system transparent to the local work station 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 Sharing a Pool with Multiple Interactive Subsystems: Multiple interactive subsystems in a pool speed up sign-on and sign-off by running in parallel and reduce the impact of the line recovery.

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 are excessive CPU and disk processing functions on the AS/400. Consider using the DSCJOB command function. This will suspend the job and re-display the sign-on prompt screen. 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 Work Station Errors" on page 161 for information on automatically terminating a disconnected job via 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 could generate a large number of I/Os by habitually looking for all the J′s or S′s rather than JOHN* or SCHM* .

If you must use generic searches, use a ″starts with″ (for example ABCdefgh...) technique, rather than an ″is contained in″ (for example abcDEFgh...) technique.

Increase Storage Pool Size/Lower Activity Level: This just restates the need to do good basic tuning to control paging rates.

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 via the Add Prestart Job Entry command. For programming considerations refer to the *AS/400 APPC Programmer's Guide*, SC41-8189 and for SAA CPI-C the *Common Programming Interface - Communications Reference*, SC26-4399.

See also 10.6.3, "Using PreStarted Job Support" on page 234 for additional considerations/

Use 5250-Type Workstations Rather than ASCII or 3270-Type Devices: ASCII displays and 3270 displays require the ASCII Work Station Controller or the system (3270 displays) 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 under 3270 Remote Attachment support. Default processing on the AS/400 is 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 will vary, 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. PTFs are available on various OS/400 releases that omit the 5250 data compression. Contact your service provider for appropriate PTFs as there are several sets of PTFs, depending on whether standard 3270 remote controllers, TCP/IP TELNET, 3270 DHCF, etc. are being used. You may use the following APAR numbers when discussing the need for the support with your service provider:

o MA09570: Network Routing Facility, SNA Primary Logical Unit Support

o MA09571: TELNET

o MA09572: 3270 Distributed Host Command Facility

o MA09574: 3270 Remote Workstation Controller

Controlling 5250 data stream optimization support is included in V3R1.

With PTFs applied or base V3R1 OS/400 support, 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) will look for this data area and set 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 to determine its current data stream optimization value.

The following is an example of 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')
```

The *Performance Capabilities Reference* manual, ZC41-8166 contains more performance information on remote 3270 displays and ASCII displays.

Avoid Logging of CL Commands and Outputting Job Logs: In a production environment, the logging of control language commands during application activity and generation of 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 index entry *system value* for information on automated spool file cleanup (QRCLSPLSTG). You may also use the GO ASSIST menu to schedule clean up of job log and problem dump spool 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 4MB.

- When using AFP overlay support, group print files using the same overlay in sequence for printing.

  This enables V3R1 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.

  In pre-V3R1 releases, the overlay is sent to the printer immediately preceding each print file data, regardless of whether the printer already has the same overlay stored.

- 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.

  Please pay special attention to the contents of the DTAFMT parameter when sending files to computers on previous releases of OS/400 because all the values may not be supported.

Save and Restore: For best overall system performance always understand what other system work would be active during the save or restore, 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 proper amount of 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 3490 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 tape device and system disks attached over the same I/O bus

     Maximum 3490 rates cannot be achieved if the 3490 and system disks are attached over the same I/O bus. To achieve maximum 3490 throughput you must place the disks and the 3490 tape device on separate buses.

   - 6390 internal tape device and system disks attached to same Multi-Function I/O Processor (MFIOP)

     Restore 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 1MB of storage available will get at least a moderate level of throughput. At 6MB of available storage, a single save and restore session will achieve close to its maximum throughput. If multiple save and restore sessions are to be concurrently active, allow approximately 6MB for each concurrent save and restore session.

Beginning with V2R3 save and restore performance in a non-restricted state (save/restore and other jobs active) was improved to approaching that achievable in a restricted state. To achieve maximum throughput, the storage pool activity level to available storage should be set to a factor of 1:6000. This means, each concurrently active save or restore job (session) should have 6MB available.

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″ will give more processor power to the this job(s) and achieve improved throughput.

   However, running any ″batch type job,″ such as a save or restore, at a priority higher than an interactive job will degrade 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, mirroring, or system checksum environments

   System checksum, 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.

   As a general rule of thumb system checksum yields the worst degradation compared to no protection. In most cases it could take twice as long to restore. As a result system checksum is not recommended with the availability of the 9337-2xx and 9337-4xx disks and internal disks in RAID mode attached to the 6502 disk controller.

   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.

   Laboratory tests have been conducted with internal ″benchmarks″ for system checksum, mirroring, and RAID-5 protection. Two test results are shown in Table 11 on page 167 where the same 3490 tape and object were restored. 6502-6605 represents the Advanced Series new disks on the 6502 controller. The time to restore with no DASD protection is considered a ″1″ - the base restore time. Various DASD protection schemes are listed relative to ″1.″ For example, a value of ″1.1″ for RAID-5 means the restore took 10 percent longer.

Table 11. Restore Time Comparisons: Various DASD Protection Methods						
Data Restored	**Base 9337-0x0**	**System Checksum**	**Mirroring**	**9337-1xx**	**9337-2xx**	**6502-6506**
Source Members	1.0	2.8	1.1	1.6	0.5	0.4
Large File	1.0	4.4	1.1	2.2	1.4	1.3
Note: The 9337 models shown are 1 drawer models. Mirroring results compared shows a measurable range when compared to RAID-5 on the new DASD configurations. In general restore onto a system with mirroring can range between 0 to 10 percent slower versus a system with no DASD protection.						

For save environments system checksum and RAID-5 protection are not involved in the process. In a mirrored environment laboratory test results showed save times with mirroring can range between 3 percent faster to five percent slower when compared to no DASD protection.

System tuning should be conducting to achieve the maximum performance that can be achieved in the ″normal backup″ mode of operation.

Always consider using Save-While-Active and saving only the objects that have changed since the last save.

Additional save and restore performance enhancements beginning with V2R3 include:

- Save-While-Active support allows **multiple** library/object synchronization

  This improves save speed while saving objects that are in use by currently active applications. prior to V2R3.

- Reduced time to save and restore multiple member source files

  The storage for internal source file attributes was re-optimized, reducing the amount of space needed to save and restore source files. Improvements in laboratory test scenarios range from 11% to 70%.

  In order to achieve faster source file save and restore times the following steps must be completed:

  1. Save source files successfully onto a tape
  2. Delete the source files
  3. Restore the source files back onto the system

- 9404 system support to attach 3490 models via the 2644 I/O Processor (IOP) provides improved save and restore performance

  Assuming the 3490 costs can be justified, such as sharing with other systems, this enables 9404 customers to use the fastest tape devices supported by AS/400.

See 10.11.3.4, “LAN Server/400 (QLANSrv) Save/Restore Performance” on page 283 for save restore performance considerations for data stored in the new Integrated File System.

The *Performance Capabilities Reference* manual contains test results for many save and restore configurations.

9.13 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, etc.) and the data flow protocol (SNA, TCP) as implemented on the AS/400.

  Through V3R1, the AS/400 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

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. Some comparison of lab tests between APPC and TCP/IP batch transfer is included as these protocols received performance enhancements in V3R1. The purpose of these comparisons is to establish realistic expectations.

When examining communication performance, first analyze the line and control unit line protocol parameters, then the higher level protocol parameters such as SNA Maximum RU Length and Pacing values, and finally the application design. Application design would include 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.

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.

Review the following V3R1 manuals for additional protocol-specific information:

- *OS/400 Communications Configuration*, SC41-3401
- *OS/400 Remote Work Station Support*, SC41-3402
- *OS/400 ISDN Support*, SC41-3403
- *OS/400 Local Area Network Support*, SC41-3404
- *OS/400 X.25 Network Support*, SC41-3405
- *OS/400 Communications Management*, SC41-3406
- *OS/400 TCP/IP Configuration and Reference*, SC41-3420
- *OS/400 Sockets Programming*, SC41-3422
- *OS/400 ICF Programming*, SC41-3442
- *CPI Communications Support*, SC26-4399
- *Performance Capabilities Reference*, ZC41-8166

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. Beginning with V2R2, the OS/400 added performance collection information for SNA, SNADS, APPN, and Station Level database files and 5494 remote control unit response time information.

Note STRPFRMON RRSPTIME(*SYS or response time range values) must be specified to collect 5494 response time data.

Beginning with V3R1 the performance monitor also collects and records data from the File Server I/O Processor (FSIOP).

Table 12 on page 170 lists the communications related QAPMxxxx database files and the type of information recorded.

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

Table 12. Performance Monitor Communication Data Files

The Performance Tools/400 produces reports based on the ommunication IOP (including FSIOP) and line and 5494 remote controller response time data. No reports are produced for the SNA, SNADS, APPN, Station Level and FSIOP data.

Beginning with V2R2, the *Performance Tools/400* (Version 3 - SC41-3340) manual 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 and trends in increasing values.

A communications performance residency was completed with V2R2 level support. A document is available to authorized IBM representatives that documents the results of this residency and sample queries and is available by sending a note to RCHVMP(JCOOK) requesting the document. There are no plans to publish this document.

Both the *Work Management Guide* and the *Performance Tools/400* manuals provide a list of all communications related database files and brief field descriptions for the data collected by the OS/400 Performance Monitor.

You should not have X.25 and BSC (Binary Synchronous) lines in the same IOP if you are using any B model of AS/400 because these are old technology IOPs and do not support these two protocols simultaneously.

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 retransmissions based on communication error recovery, the better performance will be realized.

Note that in some remote line environments, poor quality lines may require shorter frames of data just to successfully transmit data without considering performance.

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 which 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, will include, but 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.

9.13.1 Communication Line Speed Considerations

The maximum speed capability of any communication line becomes important only when enough data is sent or received that line speed becomes a possible bottleneck. For interactive environments line utilization up to 50% can deliver acceptable response time, though performance will begin degrading around 40%. For batch environments line utilization as close to 100% is desirable.

Remember that line utilization is measured in bits per second (bps) as rated line speed. In most cases 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 line speed. 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.

Version 2 communication IOPs and LAN IOPs are able to process more data much faster than corresponding Version 1 hardware. The newer IOPs (2617, 2618, 2619, 2623, 2665, 2666, 2668, 6506) are necessary or ″IOP performance assistance″ in V3R1 for improved APPC and TCP/IP performance. Much of the communications management functions, such as assembling and disassembling SNA RUs within frames has been moved to these new IOPs.

9.13.2 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 will *attempt* 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 will utilize minimal system resources. However, if the values of this second level ″recovery limit″ are misconfigured, there is a potential that this second level of recovery will consume more resources than normally.

For example, a CMNRCYLMT value of (2 5) is normal and means ″attempt to recover the first level (line or control unit retry value exceeded) failure two more times within 5 minutes.″ If this second level of recovery fails twice within 5 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 like (n 0), where ″n″ is any value, then the system will attempt recovery ″forever″ because zero means there is no time limit for the second level recovery. This could cause serious system performance problems and can be stopped by varying the configuration object off.

9.13.3 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 than can be *received* before sending an acknowledgement (LANACKFRQ), and maximum number of frames than 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, the following maximum frame sizes are possible:

  – 1994 bytes for the 4 Mbps token-ring adapter (6160, 6130 IOPS - not recommended)

  – 4060 bytes for the 4/16 Mbps token-ring adapter when running 4 Mbps (2636, 6130 IOPs)

  – 8156 bytes for the 4/16 Mbps token-ring adapter when running 16 Mbps (2636 IOP)

  – 16393 bytes for the 4/16 Mbps token-ring adapter feature (IOP) 2626 or 2619 when running 16 Mbps

    Note that the 6506 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 6506 is the File Server IOP used with LAN Server/400.

  – 1496 bytes when using Ethernet LAN adapter 2617 or 2625 at 10Mbps

  – 4444 bytes when using FDDI LAN adapter 2618 at 100Mbps

- 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 case you may need to experiment with different values as discussed below:

  - When using the new LANs (2617, 2618, 2619, 6506) test with MAXLANOUT set to 6. Test to see if performance is better than LANMAXOUT(2). If still not improved over 2, test with LANMAXOUT(4). If performance is not significantly better than 2, use LANMAXOUT(2), which is the default.
  - Lab tests show very little performance improvement occurs with LANMAXOUT set to 8 or higher.
  - When using the older LAN IOPs (2636, 2626, 2617) LANMAXOUT(2) typically gives the best batch performance
  - When communicating with A PS/2 Model 50 or higher, increasing LANMAXOUT above 2 may yield improved performance, but keep MAXACKFRQ at 1 (which is the default). For personal computers with less performance use LANACKFRQ(1) and LANMAXOUT(2) as this reduces the occurrence of buffer overruns.

Note that in mixed environments LANACKFRQ(1) favors the interactive data exchange at the expense of batch transfers. LANACKFRQ could be increased to improve batch transfers, but interactive data exchange would degrade.

General considerations for the LAN IOPs include:

- The 6506 File Server I/O Processor (FSIOP) provides the fastest LAN performance with the 2618 FDDI IOP a close second. Note that the limited maximum frame size of the 2618 FDDI IOP probably accounts for the slightly slower performance.
- The 2619 TRLAN IOP provides very high performance at a lower cost than the 6506 FSIOP, which also contains file serving software.
- For Ethernet, the 2617 offers much better performance than the older 2625 IOP.

  We did not have performance results for the 6506 with the Ethernet LAN support but presume throughput is at least equivalent to the 2617 IOP.
- 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 and Resource Interval reports to observe line and IOP utilizations. The Advisor function will also highlight possible utilization threshold problems.
- In very heavy LAN data transfer environments always consider spreading workload across multiple IOPs

Table 13 on page 175 shows the best achievable LAN data transfer rates for a single program, one-way transfer for most of the supported LAN IOPs. D model AS/400s were used. Note that the D and later AS/400 model series support communication IOPs that have not had IOP utilization rate problems and are required to support the V3R1 ″IOP performance assist″ discussed later in the redbook for APPC and TCP/IP protocols. IOP feature numbers are listed in parentheses.

Table 13. AS/400 LAN Lines - Maximum Throughput Capabilities				
PROTOCOL/IOP	**Rated Line Speed**	**Frame Size**	**LANACKFRQ, LANMAXOUT values**	**Achieved Data Rate (Mbps)**
TRLAN (2636)	16Mbps	08156	1,6	02.00
TRLAN (2626)	16Mbps	16393	1,6	09.59
TRLAN (2619)	16Mbps	16393	1,6	11.67
TRLAN (6506)	16Mbps	16393	1,6	14.91
Ethernet LAN (2617)	10Mbps	01496	1,6	07.25
Ethernet LAN (2625)	10Mbps	01496	1,2	00.93
FDDI (2618)	100Mbps	04105	1,6	13.42

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. Large frame size and best values for LANACKFRQ and LANMAXOUT were used and shown (for example, 1,6 = LANACKFRQ(1), LANMAXOUT(6)). Other frame sizes or LANACKFRQ and LANMAXOUT values resulted in slower performance than shown.

The 6506 File Server IOP with Token Ring adapter provides fastest AS/400 to remote system throughput. The 6506 with Ethernet LAN adapter provides performance equivalent to the 2617 Ethernet adapter.

AS/400 to AS/400 programs using ICF interfaces, APPC and *CALC for SNA MAXLENRU with 32,763 byte records were used to minimize the inter program processing within OS/400 and Licensed Internal Code (LIC).

Most customer applications will realize lower rates than those shown as they will typically perform many database I/O operations and use significantly shorter user program record lengths.

9.13.4 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 under 9.13.3, “LAN Line and Control Unit Parameters” on page 173.

AS/400 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, the setting of these values can be quite flexible. In cases where the AS/400 is communicating with a remote workstation controller, such as the 5394 or 5494, the controller will support smaller maximum values than the AS/400.

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 *Communications Management Guide* has a thorough explanation of how these parameters interact and provides some examples. In general, consider the following:

- Use the configuration defaults for most parameters until the remote communications environment is understood. The line type protocols are only 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 via device file parameters (such as RSTDSP and DFRWRT) and program output operations can sometimes have greater impact than the configuration parameters.

- For SDLC multipoint lines at speeds of 9600 bps or higher, where it is typical for more than 5 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 3 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 would only wait a maximum of one-half second for each control unit or experience 3.5 seconds of inactivity for 7 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 would cause the system to poll this control unit approximately every 90 seconds. Use of 90 seconds does have a disadvantage for the control unit when it is powered on. This is because it could 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 column ″Pct Poll Retry Time″ on the Resource Interval report - Communication Line Detail page shown in Chapter 7, "Performance Problem Analysis." 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. 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 will be sent and then the next control unit will be communicated with.

The typical scenario for using 1 or 2 for OUTLIMIT is when seven frames to a control unit results in partial screens 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 screens. OUTLIMIT(1) or OUTLIMIT(2) would enable 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 misuse of OUTLIMIT could permit a specific control unit to monopolize the available line capacity.

9.13.5 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 include the following. (You will note many are simlar to those listed under 9.13.3, “LAN Line and Control Unit Parameters” on page 173.)

- Data rate (throughput) improves as the frame size is increased.
- The newer IOPs (2617, 2618, 2619, 2623, 2665, 2666, 2668, 6506) provide the IOP assist 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 V3R1. A key packet network condition that should be reviewed is *packet congestion*. Congestion occurs when either the AS/400 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 or received by the AS/400.

  See Appendix H, “Sample X.25 Queries for Network Congestion” on page 481 for sample queries to determine if congestion is a cause for poor X.25 performance. The X.25 receive not ready counts are not reported on the Resource Report and require a user query to determine if congestion is the problem.

- Given the same single batch data transfer application, CPU models, and SNA MAXLENRU and Pacing values a Frame Relay line delivers the maximum throughput rate.
- Asynchronous and Binary Synchronous line protocols are significantly less efficient than the other protocols.
- For interactive environments, line utilization at 50% or less give good performance.
- 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, 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, etc.), look at the communications line congestion values under the ″Congestion Local - Remote″ heading for high send or receive values.

  See Figure 100 on page 400, Figure 101 on page 401, Figure 102 on page 401, and Figure 103 on page 401 for sample output. Consider the Performance Monitor time interval to determine if the local values are excessive.

  If the AS/400 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 14 on page 179 shows the best achievable WAN data transfer rates for a single program, one-way transfer for most of the supported LAN IOPs. D model AS/400s were used. Note that the D and later AS/400 model series support communication IOPs that have not had IOP utilization rate problems. IOP feature numbers are listed in parentheses.

Table 14. AS/400 WAN Lines - Maximum Throughput Capabilities				
PROTOCOL/IOP	**Rated Line Speed**	**Frame Size**	**LANACKFRQ, LANMAXOUT values**	**Achieved Data Rate (kbps)**
SDLC (2623)	56kbps	2057	na	54
SDLC-fT1 (2626)	640kbps	2057	na	541
Frame Relay (2666)	2048kbps	8182	1,2	1864
X.25 (2623)	56kbps	4096	na	55
ISDN/IDLC (2623)	64kbps	8192	na	63
ISDN/X.25 (2623)	64kbps	4096	na	62

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 are shown.

Other frame sizes or LANACKFRQ and LANMAXOUT or MAXLENRU and PACING resulted in slower performance than shown.

AS/400 to AS/400 programs using ICF interfaces, APPC and *CALC for SNA MAXLENRU with 2048 byte records were used to minimize the inter program processing within OS/400 and Licensed Internal Code (LIC).

″fT1″ is over a fractional T1 link.

Most customer applications will realize lower rates than those shown as they will typically perform many database I/O operations and use significantly shorter user program record lengths.

9.13.6 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 will be 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. Class of Service priority is discussed at the end of this SNA Device Parameters section.

- On the AS/400, 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 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. With Version 2 Release 1, the AS/400 supports a maximum RU size of 16384 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 APPC/APPN devices, the Mode Description controls MAXLENRU values. For other device types such as 5250 displays the AS/400 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 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 via 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 running Remote Job Entry (RJEF), 3270 Device Emulation, Distributed System Node Executive (DSNX), and Distributed Host Command Facility (DHCF).

Refer to the *AS/400 Communications Management Guide* 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 will be sent. A high pacing value will improve performance for an individual LU.

  For APPC/APPN devices, the mode description controls PACING values. For other device types the AS/400 takes a default based on device type or pacing can be explicitly specified on the device description. In most cases a pacing of 7 (seven RUs sent before waiting for a positive acknowledgement) is the default. Release 2.2 microcode for the 5394 is required in order to use a pacing value greater than 3 for an attached printer.

  Most display devices run with a pacing value of 0 (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 displayed on each screen.

  Batch type communications such as file transfer or remote printer output would also normally choose a large pacing value for maximum throughput.

  When the AS/400 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 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, screen 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 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, then 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 will 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.

9.13.7 Printers Attached to 5394 or 5494 Controllers

5394 microcode diskette Release 2.2 and Version 1 Release 3 OS/400 or later support 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 AS/400 and the workstation controllers must be appropriately configured to take advantage of the larger values. On the AS/400, the appropriate line description must specify 521 or larger as the MAXFRAME size. The workstation controller description must specify 521 or *LINKTYPE for MAXFRAME size. The printer device description should specify MAXLENRU(*CALC) and the PACING parameter should be set between 3 and 7 (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 3. 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.

The *Performance Capabilities Reference* manual discusses printer performance more thoroughly.

Note that over time new models or features of the 5494 will be introduced. Watch for larger frame sizes and make use of them as they become available. Also note that APPC mode QRMTWSC is provided specifically for the 5494 and should be used for MAXLENRU and PACING values.

9.13.8 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 the AS/400 communications support, CPU utilization will be a factor that needs to be considered against the entire operating environment.

AS/400 supports data compression for:

- File Transfer Subroutines (FTS) which 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 via 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/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/decompression will be used.

- SNA DSNX support

  The host system Distributed System Executive support user selects whether compression/decompression will be used.

- SystemView Managed System Services/400 (MSS/400)

  Managed System Services support provides change management functions based on the managing system parameters. V2R3 MSS/400 provided a superset of DSNX change management functions when managed by NetView Distribution Manager V1R5 or later.

With V3R1, MSS/400 provides some enhanced change management functions on the local AS/400 and can be managed either by V3R1 SystemView System Manager/400 or by a host Netview Distribution Manager.

For V3R1, MSS/400 may take advantage of the APPC compression support discussed under ″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)

  For V3R1 System Manager/400 gains the capabilities to perform change management support very 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 running MSS/400 or a RiscSystem/600 or OS/2 system running a NetView Distribution Manager ″client″ or ″catcher″ program.

  For V3R1, SM/400 may take advantage of the APPC compression support discussed under ″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

  Beginning with V2R3, the AS/400 *mode description* supports several data compression/decompression algorithm options via 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- 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-, 10-, and 12-bit codes.

In general, CPU utilization caused by compress/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.

9.13.9 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, 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 very good.

If data is transferred primarily in only one direction the advantage of full-duplex support is minimized because most half-duplex modems currently available have very short turnaround delays.

For concurrent batch and interactive on a LAN line, refer to LAN parameter values shown under the 9.13.3, "LAN Line and Control Unit Parameters" section.

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 will give 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 7 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 7 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.

9.13.10 General V3R1 APPC and TCP/IP Performance Expectations

Prior to V3R1 APPC batch file transfer was generally considered acceptable, but with the increasing importance of client server file serving and database serving performance, APPC performance needed to be improved.

Prior to V3R1 TCP/IP batch file transfer was generally considered unacceptable when compared to OEM implementations.

In V3R1 both APPC and TCP/IP code paths have been significantly shortened and receive ″IOP Assist″ when the ″faster IOPs″ (2623 WAN IOP, 2617 Ethernet IOP, 2619 Token Ring IOP, 6506 FSIOP, 2618 FDDI IOP, 2666 Frame Relay IOP, 2665 SDDI IOP, 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.

9.13.10.1 V3R1 APPC Performance Improvement Summary

For APPC applications the decrease in CPU seconds was the most notable performance improvement. Interactive response time was also improved and batch file transfer rates improvements ranged from little up to 20% Improved CPU seconds lets the same CPU model run more APPC-based applications concurrently than V2R3 with the same CPU impact.

APPC file transfer CPU seconds were reduced by approximately 50% (2 times faster) for small record lengths and up to 66% (3 times faster) for very large application record lengths (eg 32,763 bytes) compared to V2R3.

9.13.10.2 V3R1 TCP/IP Performance Improvement Summary

For V3R1 both CPU seconds and data transfer rates have improved significantly compared to V2R3. Specific highlights include:

- When the TCP/IP application uses the new V3R1 sockets interface rather than the V2R3 PASCAL API, CPU utilization was reduced from 4 to 9 times and TCP/IP data throughput rates were improved by 4 to 7 times.

  Performance improvements are best for the larger application record sizes.

- V3R1 TC/IP FTP (File Transfer Protocol) data rates improved from 3.5 to 4.7 times compared to V2R3.

  For optimum FTP performance the AS/400 target file should be allocated prior to doing the file transfer (Get or Put operations).

- Exchange of ASCII data is slower because of the need to translate between EBCDIC and ASCII prior to data transmission.

- TCP/IP performance is still not as fast as APPC for data transfer but the performance is much closer

  With the V3R1 improved APPC performance, V3R1 TCP/IP bulk data transfer rates for the same record lengths and frame sizes are now approximately 50% slower than APPC, except for some specific application environments when the remote system has a very high processor speed, such as a RISCSystem/6000 performing a backup to ADSM/400. In at least one test result TCP/IP and APPC throughput were very close.

  See 10.15, "ADSTAR Distributed Storage Manager/400 Performance Tips" on page 308 for sample test results.

- V3R1 AnyNet support enables APPC/SNA over TCP/IP networks and TCP/IP over SNA networks but AnyNet degrades ″normal protocol″ performance.

  The use of AnyNet/400 as part of OS/400 is enabled through the Network Attributes (ALWANYNET(*YES)). Additional AS/400 CPU is required to perform the ″cross-protocol″ translation within the AS/400. ICF and CPI-C application program interfaces are best for APPC and Socket interfaces perform best for TCP/IP, each over their native protocols stacks (APPC/SNA or TCP/IP links).

  When ALWANYNET(*YES) is specified, all socket applications will will experience slightly longer code paths. This is due to system checking to determine if the sockets data must use AnyNet support.

9.13.11 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 8 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 could 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 netork 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 occurs:

1. Eventually all 12 of the switched lines will be in use simultaneously. TRS will receive 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 via 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 the TG is

active but not in use. Also, since there are now lines available for the other 103 controllers, TRS is told via another 103 TG Updates that each of those TGs is now active but not in use.

4. The lines that just became available will be used to connect to some other of the 115 remote end nodes, which means TG updates will be 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 via another 103 TG Updates that those TGs are inactive.

5. Steps 3 and 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.

For more details on how APPN works, refer to *AS/400 APPN Problem Management*, GG24-4222.

Chapter 10. Design and Coding Tips

In addition to the tips discussed in Chapter 9, “System Performance Tuning Tips” on page 131, 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
- 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

At the publication date of this redbook, there was no feedback from customer on the use of ILE constructs within actual applications or Client Access/400 for WINDOWS 3.1 clients or LAN Server/400. Therefore the tips and considerations discussed herein are limited to development and Rochester ITSO residency experiences.

10.1 General Application Programming Tips

Use Odd Length Packed Fields for Numeric Data: The AS/400 does packed decimal arithmetic. Extra IMPI 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 will improve 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 will be 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 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 V3R1 enable passing data in ″constructs″ unique to UNIX on other systems. Complete discussion of UNIX capabilities is beyond the scope of this redbook.

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, 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 will have degraded performance if frequent decimal data errors are signalled.

Decimal Data Performance: For RPG and COBOL use packed decimal data for best performance.

Binary Data Performance: For C/400 use binary integer data for best performance.

Use Fortran or C for Calculations: These languages can provide a performance boost for high use programs which 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.

Beginning with Version 2 Release 1 this same ″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 will be incurred for each call.

Note that in V3R1 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. 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.

With V3R1 the AS/400 offers this kind of program development 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 3.10.5, "Activation Groups and the Integrated Language Environment" on page 46.

Note that performance testing with V3R1 indicates RPG and COBOL ″already resolved system pointer″ support for the second and subsequent **dynamic call** *processing is significantly slower than a a* **call bound** *program within the Integrated Language Environment support.*

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 3.10.5, "Activation Groups and the Integrated Language Environment" on page 46, 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 and, as such will impact 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 recommend 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.

ILE C/400 Create Bound C (CRTBNDC) command supports ACTGRP(*NEW), but not *CALLER. For C you must create a module (CRTCMOD command) and then 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.

We recommend keeping the number of procedures within a service program to a reasonable number. The static storage for all 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, then consider the following approach as an alternative to the normal passing of parameters. The technique will use 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 that 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.

V3R1 provides the Reclaim Activation Group (RCLACTGRP) command to clean up all resources assigned to a named activation group.

Be carefull when using this command. It is recommended to not 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 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 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 data-base operations within the program. For interactive transactions, consider 20 database operations as an excessive number.

Consider Program Observability: Using RMVOBS(*ALL) on the CHGPGM, CHGSRVPGM, and CHGMOD commands can release a significant amount of disk space, which only 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* via the PGMOPT parameter - CPROBJ(*OBS).

Caution Against Removing Observability

If the application programs are to be moved to the AS/400 Advanced Series that uses the PowerPC-based technology (extended for AS/400), they must have observability or the ″restore″ will fail. 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 then return to a primary or ″driver″ program. In many cases the called programs repetitively use the same database files and display screen record formats.

In this scenario the primary program should open the most frequently used database files with the maximum processing possibilities (read, write, update) with SHARE(*YES) on the files. The secondary program open file operations will connect with the existing Open Data Path (ODP) which will result in significant CPU savings and response time improvement.

All the display formats used by the primary and secondary programs can be placed into a single display file that can then be opened by the primary program with SHARE(*YES). This will speed up the file open by the secondary programs and can be used to permit screen data to be retained across calls to several programs.

There are some cautions when using shared database and display files. The secondary programs must be written with the understanding that database file record (cursor) positioning and workstation screen 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 then call program PGM2. PGM2 uses shared file SHARE and causes the file to be positioned to record 23 and then 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 what 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 could result in large Process Access Group (PAG) sizes which could result in degraded performance on storage constrained systems where many jobs are using the same display file.

Look at 8.1.1, “Comparing Coding Techniques: An Example” on page 112 for an example of CPU consumption using SHARE(*NO) and SHARE(*YES).

If 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 I/P 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 as filling a subfile page), keeping the ″DDM I/O operations″ per interactive transaction below 6 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 will be more noticeable.

Increase Code Sharing: The sharing of a single copy of a program by all active users, is built into the AS/400. (All AS/400 programs are reentrant.) 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 would be 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 will create 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. (Note Version 2 Virtual Terminal Application Program Interface (API) is also available. It would not normally be in the list of choices. However, review of this API should be considered. See the *System Programmer's Communications Interface Guide*, SC41-0027, for more information.)

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 are 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 interfaces described in *System Programmer's Interface Reference*, SC41-8223.

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 will start to degrade.

  Entries are sent to and received from data queues and the entry remains on the queue even after it has been received. 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 is cleared or deleted and re-created.

  Note that in March of 1995 OS/400 PTFs became available to support remote data queue and data area access via DDM support. These PTFs are valid for V2R3, V3R0M5 and V3R1M0.

  – Summary of data queue support via DDM

    The data queue APIs QCLRDTAQ (clear data queue), QSNDDTAQ (send to data queue), QRCVDTAQ (receive data from data queue), and and the CRTDTAQ (Create Data Queue) and DLTDTAQ (Delete Data Queue commands support remote data queues via DDM.

    CRTDTAQ identifies the remote location and the remote data queue.

  – Summary of data area support via DDM

    The data area OS/400 commands CRTDTAARA (Create Data Area) and CHGDTAARA (Change Data Area) and the API QWCRDTAA (retrieve data area) support remote data areas via DDM.

    CRTDTAATA identifies the remote location and the remote data area.

    The RPG data area functions are support also.

  Note that this DDM support is in English Upper Case only. The following is the initial list of DDM data queue and data area support PTFs. The PTFs also provide information on use of English Upper Case support when it is not the primary language and the sequence of installing these PTFs if OptiConnect/400 is also installed.

```
                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 will temporarily diskable OptiConnect400 DDM.
               After installing this PTF. Apply B968529, if and only
               if you are using OptiConnect/400.
     B968529   This PTF will restore 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: Several applications convert the MMDDYY format into YYMMDD format by multiplying the MMDDYY format by 10000.01. This can take up to 40 times more CPU than some 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: The Performance Tools/400 helps you to 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.

For analysis of individual program performance, the Performance Tools Sampled Address Monitor support can show the relative amount of processing time spent in parts of a *program or set of programs*. You cannot identify the total program processing time, but you can determine what areas of a program are most frequently used.

A histogram will be produced to identify the statements which are in control during a time period. The following example shows how to start the Sample Address Monitoring.

```
STRSAM library/program identifies the program to have instruction
 monitoring started for

STRSAMCOL begins data collection
```

Programming PRPQ Timing and Paging Statistics (PRPQ P84204) can identify high CPU and disk utilization programs, system modules and system tasks.

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 the Operating System/400, such as Open Query File (OPNQRYF) and the SAA Query Common Programming Interface (CPI) known as Query Management/400 CPI. Other query facilities are separately-priced licensed programs, such as Structured Query Language (SQL), program number 5763-ST1, and Query/400, program number 5763-QU1. Note that starting with V2R2, SQL provides SQL/400 Query Manager, an interactive interface to the capabilities provided by Query Management/400 CPI.

Look out for any generic searches and multiple format logical files, as those are big hitters for performance.

In general, writing a program that must individually process a small set of records will be more efficient than use of 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, use of the query interfaces should be considered.

Comparison of user programming versus the various query facilities is beyond the scope of this publication. However, 10.3, “Queries and Structured Query Language (SQL)” on page 214 provides more query-related performance tips.

10.1.1 Application Error Handling

Improper handing of error conditions within a program can impact system performance it the error condition causes a program (job) to loop or flood message logs or the job log with messages. Consider the following:

- Errors happen frequently.
- If the application does not handle them, the system will sometimes detect a workstation error loop and end 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 will end the activation group. The next ILE call using that activation group incurs the overhead of starting the activation group. See index entry for *activation groups* for additional information.

- 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′s absolutely necessary.

See "Use DEVRCYACN or Program for Work Station Errors" on page 161 in Chapter 9, "System Performance Tuning Tips" on page 131 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 could have been avoided if this additional effort was incorporated into the application design and implementation.

What happens in an application that doesn′t 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 from where he/she was before the error. What does this scenario cost in performance? More disk I/O and CPU and memory are consumed than would be 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 screen, and then create another job when the sign-on data is entered.

Errors are detected at all levels of the system (LIC, 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 LIC log. Other errors, such as 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 LIC error logging task.

10.2 Database Tips

This section begins with general database usage performance tips and follows with query-based performance tips. Note that beginning with V3R1 database communicates ″no record found″ and ″end of file″ via parameters rather than via MI exceptions as done in all previous releases. 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), the following lists the processing techniques in rank from best performance to worst performance:

1. Sequential processing (blocking can be used to maximum advantage)
2. Random by relative record number
3. Sequential by index (key)
4. Random by index (key)

SEQONLY and NBRRCDS for Sequential Processing: The Override Data Base File (OVRDBF) and Open Data Base File (OPNDBF) commands support blocking parameters for Sequential Only (SEQONLY). The OVRDBF command supports the NBRRCDS parameter which provides some different blocking support which will improve processing in some cases. These parameters can significantly improve sequential processing performance. They enhance blocking of 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 number of records is recommended since SEQONLY(*YES) defaults to a relatively small buffer of 4KB which minimizes advantages of blocking for large records. 32767 is the maximum number of records that can be specified.

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 *AS/400 Control Language Reference*, SC41-0030, has a thorough discussion of the sequential only number of records support under the Override Database File (OVRDBF) command.

Distributed Data Management between AS/400s will use the number of records value in SEQONLY to determine the frequency of ″DDM function acknowledgements.″ For example, assuming SEQONLY(*YES 200) were specified on the OVRDBF command that references the DDM file, then a communication line trace shows that after records 2601 through 2800 (every 200 records) have been 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 OVRDBF command NBRRCDS 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. Too large a value will degrade the performance.

Through Version 3 Release 1, the largest physical block of database data that can be exchanged with the disk is 32,767 bytes. So, 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 OVRDBF command NBRRCDS parameter can be used to get the Licensed Internal Code to bring in a specific number of records from the disk when either sequential or random processing will be 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, then blocks of records read from the disk will go unprocessed. If this occurs frequently, the system will suffer 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 only one physical file and the physical data is organized in key sequence. This will enable many records to be read into main storage that will be 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, the LIC 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 will severely degrade performance. Use the system default for NBRRCDS in this case. If you have sufficient main storage, consider also SETOBJACC command support for either random or sequentially processed files or expert cache for sequentially processed files. See the index for other information on SETOBJACC and expert cache support.

Use Relative Record Processing for Random Processing: Such processing can be done on the AS/400 and it is very efficient. Use it when you have a choice of ways to access the data.

Do Not Use Low Force Write Ratio Values: Force write ratios 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 FRCRATIO value of 1 severely degrades application performance.

Note if all of your data is placed on disks that have controllers (6502, 6501) with disk write caching support, low FRCRATIO values may result in acceptable performance.

Use Join When Processing Multiple Files: A join file brings records from multiple physical files together more efficiently than an application program doing multiple reads to those same files. Note that 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.

Understand the application needs as there will be cases where the join range of function will not be sufficient.

For additional join file tips, see 10.3, "Queries and Structured Query Language (SQL)" on page 214.

Use 2nd Normal Form (as Opposed to 3rd Normal Form): Normalization is a database design technique which 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 the degree of normalization, the greater the number of physical files which have to be processed by the application.

Share Access Paths: Access path (index) sharing is a natural function on the AS/400. That is, when a logical file is created, the system will check to see whether an existing access path will meet the needs of the new file. Heavy use of logical files for select/omit processing will interfere with this sharing. An access path which 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 only critical access paths as update *IMMED. Use *DLY or *REBLD for other access paths.

Share Open Data Paths: The sharing of database files reduces CPU processing. See 10.1, "General Application Programming Tips" on page 189 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 has an exclusive library (directory) that can be used to hold temporary objects such as files that can be used by only that job. If you are coming from a System/36 and do a lot of file creations (BLDF), you will get better performance if you create these files once at the start of the job and use the CLRPFM command to clear the file(s) rather than creating and deleting the files being used. One of the advantages of QTEMP is the fact that the system will clean up and delete all objects created into QTEMP when the job terminates (SIGNOFF).

Beginning with Version 2, the System/36 environment has a special cache-extension option to this library for System/36 environment jobs that will 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-, write-, and update-capable. This only works in the System/36 environment.

Consider When to Use Open Query File: In batch report job streams, run 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 only 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.

Use CPYF Instead of CRTDUPOBJ: If your (batch) program performance is deteriorating even though there is no significant increase in file sizes you should consider using CPYF on the files being processed. A possible reason is that the files tend to get disorganized.

This can cause a noticeable improvement in the run time.

Minimize the Number of Logical Files: Do not create many logical files over the same physical file with very 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 related information on sharing access paths.

Minimize the Key Size in Logical Files: The larger the key the higher the CPU usage resulting from more frequent physical I/O.

Review existing logical files. If dropping the right most key field does not change the order, then drop that field from the key.

Anticipate Additions to Files: For files which are likely to exceed the default size of 10,000 records and three extents of 1,000 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.

Consider Limiting Reuse of Deleted Records: The AS/400 normally does not reuse the disk storage that contains a record that was deleted when adding a new record. To free up the storage (space) taken up by deleted records, the standard technique is to reorganize the file with the RGZPFM command during some off-peak time period. This continues to be the best performing environment for applications heavily adding new records.

The physical file REUSEDLT(*YES) parameter tells the system to attempt to place a new (added) record into a space that currently contains a deleted record. Reuse of deleted record space can significantly reduce storage consumption in those environments where it is difficult to allocate the time needed to reorganize a file (RGZPFM command).

However, there is system overhead in identifying the deleted record space when adding a record to the file. This overhead is not significant when reading records or adding a few records per unit of performance measurement, such as interactive response time. As the number of record adds to deleted record space increases per unit of time, the performance degradation versus REUSEDLT(*NO) can become significant until deleted record space can no longer be used. When the deleted record space can no longer be used, the added records are processed as if REUSEDLT(*NO) were specified for the file. Any performance degradation would be most noticeable in a batch environment.

The performance gain of adding records to deleted record space is best when there are contiguous storage locations containing deleted records. This minimizes system overhead needed to determine where to place an added record. One way to ″know″ you have a large contiguous space of deleted records, is to begin processing a file with no records or a file immediately after a file reorganization. If the application processing adds new records and then deletes all of the just added records (such as in a file used to log new orders), there should be a contiguous space containing the deleted records.

The *Performance Capabilities Reference* manual should be reviewed for specific performance test results.

SETOBJACC: 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 that 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.

For more information see:

- *Work Management Guide*, SC41-8078
- Appendix E, "OS/400 Expert Cache and Set Object Access Overview" on page 441

The *Performance Capabilities Reference* manual should be reviewed for specific performance test results.

Expert Cache: Consider using expert cache support. By specifying *CALC for a a shared storage pool, you ask the system to monitor object reference patterns within the storage pool and, if it detects 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.

Note that any deleted record space is included for a database file object.

For more information see:

- *Work Management Guide*, SC41-8078
- Appendix E, "OS/400 Expert Cache and Set Object Access Overview" on page 441

The *Performance Capabilities Reference* manual, ZC41-8166, should be reviewed for specific performance test results.

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 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.
- 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 the similar logic described above with the two file approach by reading the record first with the read-without-lock operation. Note that since the file was opened for both input and output, the read without lock will wait 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. Note that 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.

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, including 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 *Data Base Guide*, SC41-9659, discusses the possible use of dynamic select but omit in Chapter 3. However, our recommendation is to do some testing and use the approach that is best 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 and the logical file is seldom used and unique keys are not required.

Also 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: 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: V3R1 database trigger support represents one of the most powerful features of DB2/400. Triggers are user-written programs which 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 would do, 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 could result if certain jobs perform thousands of adds/deletes/updates. The triggers could 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 will terminate the activation group, close all the files and cause 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 10,000 record insert/delete/update example above, DDM files were defined for a ″program file″ and triggers were used on the remote system to to do all 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 then 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: 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 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 will prevent the parent key deletion. An exception is returned.
    - NO ACTION: 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 will prevent the parent key deletion. An exception is returned.
    - NO ACTION: 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 *DB2/400 Programming Guide*, SC41-3702, 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 will experience better performance when their foreign key fields and parent key fields have identical null attributes. In fact, the non-null field attribute will deliver the best performance.

  Ideally, your parent and foreign key fields should not 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 and/or the update rules is 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 will provide better performance since journaling and commitment control are not required.

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.

This section provides various performance-related information on journaling implementation and refers to a new for V3R1 ″Data Base Journaling Summary″ section of the Performance Tools/400 Component Report.

See also new for V3R1 System Managed Access Path Protection (SMAPP) support under "Consider System Managed Access Path Protection (SMAPP)" on page 209 as SMAPP uses special journaling support.

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.

V3R1 Commitment control received performance enhancements in ROLLBACK and COMMIT functions. V3R1 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 when more than 10 units were available in an ASP, journal receivers simply used the first 10 configured units.

V3R1 journal receivers are spread over up to the 10 fastest disks within an ASP. Performance can be improved significantly when these fast disk units or their IOP′s 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 up to 20 - 30%. An example of such devices include the 9337-2xx/4xx disks and the 6502 internal RAID controller attached disks.

The V3R1 Performance Tools/400 Component Report includes a new ″Data Base 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 under "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.

Figure 32 on page 212 is a sample of the ″Data Base Journaling Summary″ section of a V3R1 Component Report. This figure is explained in the SMAPP section as you need to understand SMAPP to evaluate some of the column heading values.

Consider System Managed Access Path Protection (SMAPP): History from AS/400s recovering from abnormal system failures has shown typically 75% of all recovery time overhead can be charged to recovering access paths (indexes) 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 data base 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.

Beginning with V3R1, OS/400 provides automated journaling of access paths under the function called System Managed Access Path Protection (SMAPP). A *target recovery time*, applicable system wide or separately for each auxiliary storage pool (ASAP) 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 will try to ensure that all required access paths are rebuilt during the specified time frame.

The user can combine their own explicit journaling with system access path journaling under SMAPP, where SMAPP will only target access paths that are not already started by the user. **Note that 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 Processor utilizations versus no journaling. Journaling also adds to auxiliary (disk) storage requirements versus no journaling.

However SMAPP is a very efficient implementation of journaling. For example, SMAPP further reduces the overheads by grouping information in the SMAPP journal in “bundles” of approximately 32K 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 SMAPP′s log.

Refer to *Backup and Recovery - Advanced Guide*, SC41-3305, 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 ASP′s. 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 could reach up to an additional 6% where there is an excessively high frequency of access path changes (record added, deleted, 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 (1 day) is supported. Any system overhead incurred by SMAPP will need 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 V3R1 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) or Display Recovery of Access Paths (DSPRCYAP) commands or a new for V3R1 ″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 bet 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 will protect 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 will potentially cause the highest run time SMAPP CPU utilization because of its ″shortest″ recovery time meaning.

Figure 32 on page 212 is an example of the ″Data Base Journal Summary″ section of the Component Report. This section contains statistics for both user specified journaling and SMAPP provided journaling.

Column heading details follow the figure.

Note that in order to get significant journal bundle/deposit counts, we set the System Managed Access Path Protection (SMAPP) ″minimum recovery time″ (for access path recovery) to *MIN.

```
                                                          Component Report                                     11/18/94  8:23:46
  SMAPP ON - *MIN                                   Data Base Journaling Summary
                                                              SMAPP_ON
Member  . . . : SMAPP_ON    Model/Serial  . : D45     /10-A1002  Main storage . . :   80.0 MB  Started . . . . :  11/17/94 14:37:24
  Library . . : QPFRDATA    System name . . :          RCHASM02  Version/Release  :    3/ 1.0  Stopped . . . . :  11/17/94 15:02:21
       ------- Journal Operations ------   --- Journal Deposits ---   Bundle   Bundle   --Exposed AP --   --Est Exposr --
 Itv   User     User     System   System   User     System   System   Writes   Writes   System   Not      Curr     AP Not   SMAPP
 End   Starts   Stops    Starts   Stops    Total    Total    ToUser   User     System   Jrnld    Jrnld    System   Jrnld    ReTune
-----  ------   ------   ------   ------   ------   ------   ------   ------   ------   ------   ------   ------   ------   ------
14:42       4        3       45        0       36   10,491       51       48      183       42        0        1       12        1
14:47       0        1        0        0        5    8,247        0        1      165       43        0        1       24        0
14:52       0        0        9        0        0    6,870        0        1      102       42        0        0       42        0
14:57       1        1        0        0       15   12,096        0        6      175       41        0        0       51        0
15:02       0        1        0        0        2    4,610        0        1       75        2        0        0        2        0

Itv End                 --  Interval end time (hour and minute)
User Starts             --  Start journal operations initiated by user
User Stops              --  Stop journal operations initiated by user
System Starts           --  Start journal operations initiated by system
System Stops            --  Stop journal operations initiated by system
User Total              --  Journal deposits resulting from user journaled
                              objects
System Total            --  Journal deposits resulting from system journaled
                              objects (total)
System ToUser           --  Journal deposits resulting from system journaled
                              objects to user created journals
Bundle Writes User      --  Bundle writes to user created journals
Bundle Writes System    --  Bundle writes to internal system journals
Exposed AP System Jrnld --  Exposed access paths currently being journaled
                              by the system
Exposed AP Not Jrnld    --  Exposed access paths currently not being journaled
Est Exposr Curr System  --  System estimated access path recovery time
                              exposure in minutes
Est Exposr AP Not Jrnld --  System estimated access path recovery time
                              exposure in minutes if no access
                              paths were being journaled by the system
SMAPP ReTune            --  System Managed Access Path Protection tuning
                              adjustments
```

Figure 32. Performance Tools/400 - Component Report Database Journaling

Listed below is some column heading information in addition to that contained on the report page itself:

- Journal Operations: represents the number of start/stop *user-specified journal file or access path operations* and the *system journaled access path operations under SMAPP*. If the count is zero there were no start or stop operations. In some cases a start or stop issued just before a Performance Monitor time interval will appear in the next interval.
- Journal Deposits - User Total: represents the journal deposits for user specified journaled physical files and access paths.
- Journal Deposits - System Total: represents the journal deposits of access paths under SMAPP support. These deposits include those written to internal system journals and user journals if the associated files were also being journaled.
- Journal Deposits - System ToUser: represents the journal deposits of access paths under SMAPP support that were written to user journals because the user specified journaling for the associated file.
- Bundle Writes User: represents actual bundle writes (containing multiple deposits) written to user journals. These bundles include any physical file and access paths explicitly journaled by the user.
- Bundle Writes System: represents actual bundle writes (containing SMAPP-generated access path deposits) written to internal system journals.
- Exposed AP System Jrnld: represents the number of access paths in use that are being journaled only by SMAPP support (journaled by the system). This

column contains values only when the SMAPP recovery time value is set to *MIN or a recovery time value.

- Exposed AP Not Jrnld: represents the number of access paths in use that are not being journaled. This column contains values only when the SMAPP recovery time value is set to *NONE.
- Est Exposr Curr System: represents the system estimated access path recovery time exposure in minutes. SMAPP recovery time must be set to a value other than *OFF to have a value in this column.
- Est Exposr AP Not Jrnld: represents the system estimated access path recovery time exposure in minutes if no access paths were being journaled by the system. SMAPP recovery time value must be set to a value other than *OFF to have a value in this column.
- SMAPP ReTune: represents the number of times SMAPP support had to adjust its algorithm for system journaled access paths. For example, an access path that was undergoing a large number of changes now has little or no activity or an inactive access path undergoes a period of heavy changes. SMAPP recovery time value must be set to a value other than *OFF. Normally a value 0-2 is observed in this column.

V3R1 One Time Cross Reference File Conversion: As discussed in 9.2.6, "Database System Cross Reference Job (QDBSRVXR)" on page 133, installation of V3R1 requires a one time conversion of previous release database file reference database files to new files and add new information. In addition, a restore of a previous release SQL collection will require this conversion process. System job QDBSRVXR actually performs the conversion with some assistance from a QDBSRVnn system job.

Note: As shipped with V3R1, job QDBSRVXR runs at priority 0 and the QDBSRVnn jobs run at priority 9. If these jobs are busy converting reference information during normal production mode, the production jobs may experience performance degradation.

The post-install portion of the conversion may cause certain system functions which 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 V3R1 install or catalog restore or a RCLSTG or a database reclaim request (program QDBRCLXR) when the reference file conversion has not yet processed all files/tables.

You can force the conversion to complete by placing the system into restricted state after V3R1 IPL and calling V3R1 program QDBRCLXR (database cross reference reclaim) as follows:

```
QSYS/CALL QSYS/QDBRCLXR
```

This cross reference information conversion was made to enable faster access from SQL catalog files and to meet SQL standards (refer to the V3 Memo to Users).

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 V3R1M0. Starting in V3R1M0 the four additional files QADBIFLD, QADBKFLD, QADBCCST, and QADBFCST will also exist. These files are used as based-on physical files for SQL VIEW files which are created for the SQL CATALOG.

The full description of this cross reference information conversion is contained in an ″information only″ APAR II08311.

10.3 Queries and Structured Query Language (SQL)

V3R1 query support (SQL, Query/400, Open Query File support, etc.) 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 a recreate of 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 detail 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.

Before discussing tips, your attention is called to the following AS/400 documents that contain very necessary query performance information:

- For V2R2 and V2R3, the redbook *SQL/400: A Guide for Implementation*, GG24-3321.

  This redbook is very helpful and applies to many SQL applications but has not been updated after Version 2 Release 2.

- For V3R1 *AS/400 DB2/400 Database Programming Version 3*, SC41-3701.

  Appendix D, ″Design Guidelines for Query Performance.″ contains very complete information including query optimizer algorithms, join file index considerations, optimizer messages, and use of the new V3R1 Print SQL Information (PRTSQLINF) command and the Change Query Attributes (CHGQRYA) command parameters.

- For V3R1 *AS/400 DB2/400 SQL Programming Version 3*, SC41-3611

  This book and the Version 3 *Database Programming* contain redbook GG24-3321 information updated to V3R1 capabilities.

- For V3R1 *AS/400 DB2/400 SQL Reference Version 3*, SC41-3612

Generally, if the query processes a single file of 10,000 records or less, performance has been acceptable to customers. 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. However, 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:

- Print SQL Information (PRTSQLINF) command

  This command prints the access plan determined at SQL program or package creation. You can examine what indexes will be used or if it has already been decided that temporary indexes will 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(s)/table(s) and determine 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 in V2R3 or later will 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 will be issued by the optimizer if the optimizer estimates that running the query will take longer than the specified time limit. Message CPA4259 is issued that indicates the access plan that will be used. CPA4259 requires a a cancel or ignore response that can be responded to by an operator or automatically via 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.

  - `ADDRPYLE SEQNBR(56) MSGID(CPA4259) CMPDTA(QPADEV0011 27) RPY(C)`

  - `ADDRPYLE SEQNBR(57) MSGID(CPA4259) CMPDTA(USRIDNAM   51) RPY(C)`

If you do not think the query selection information will result in acceptable performance, additional detail 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.

10.3.1 Query Data Access Methods

Getting the query component to select the ″best″ data access method for your specific data organization(s) and the number of records that will actually be retrieved is important and has the following general considerations shown in table Table 15, taken from the *Version 3 AS/400 DB2/400 Database Programming* manual:

Table 15. 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.

10.3.2 General Query Tips and Techniques

This section lists several query tips and techniques. Refer to the publications listed at the beginning of 10.3, "Queries and Structured Query Language (SQL)" on page 214 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 will be closed when the program ends. It is recommended that a higher level main program be designed to call lower level programs be 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 10.3, "Queries and Structured Query Language (SQL)" on page 214 section. The key range estimate is a process the optimizer uses to estimate the number of records/rows that will be selected based on the individual predicate. In the example shown below, this would be ″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 will contain 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 then 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 will select all records/rows. However, without a primary key index the optimizer will guess only 25% of the records/rows will be selected.

The best solution in this case would be to not build this useless predicate. Alternatively ensure there is a primary key index on X so the optimizer can determine that it will select 100% of the records/rows.

Caution: Customers with large databases (over 1 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: Specifing 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 very 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.

For releases previous to V3R1, refer to PTFs for APAR SA36566. The fix eliminates a size restriction for internal sorts.

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 *NONE, *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) will allow 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 would force a temporary result file being generated, which involves more overhead in achieving the query results.
- Avoid joining two files without a JFLD or QRYSLT clause.

This includes not specifying JDDFTRCD(*YES) on the OPNQRYF command, which could result in a large number of 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 will be 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) will bias the optimizer towards builiding 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 selecting 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 will 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 will cause 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 will be 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 for at least 2MB since it takes up to 1MB per input stream task. If the disk space extent is less than 1B, 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.

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 *DB2/400 SQL Programming Version 3***, SC41-3611.**

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 1 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 Under V3R1: Recreate all SQL applications under V3R1 to reduce intermediate internal calls to the QSQROUTE and other SQL run time routines.

Consider Using an SQL Stored Procedure: Beginning with V3R1, SQL/400 supports Stored Procedures. A stored procedure is basically an already created program that is accessed either locally or remotely via SQL syntax. While the use of a stored procedure may be used locally, it is more typically thought of as a very 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, which 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 HLL such as COBOL or RPG, or in a REXX procedure, the stored procedure may perform complex computations, and access database data via either SQL syntax or AS/400 native file I/O interfaces. Figure 33 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 33. SQL/400 Stored Procedure Example

Figure 33 shows the SQL CALL statement for the stored procedure (ORDPRC) which passes values in parameters ORDNO and TOTAMT each time the procedure is called.

The DECLARE PROCEDURE statement is an optional statement that identifies the number and type of parameters that will be passed on a subsequent call, the external name of the program (the name on the AS/400), and the language of the program or procedure on the remote system (″server″). Note that the parameters will be converted as necessary, including converting data types, if required, for different languages. Use of the optional DECLARE PROCEDURE statement may improve performance of the application because it reduces some of the work that would ordinarily be 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 above 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 will not be 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 of I/O, stored procedures, and for Client Access/400 APIs using ″combined operations″, may be used to 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 V3R1 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 data bases)
- 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 like 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, as 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.

10.4 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 V2R3 *Guide to Programming Application and Help Displays*, SC41-0011, and V3R1 *Application Display Programming - Version 3*, SC41-3715. See 10.1, “General Application Programming Tips” on page 189 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. The *Version 2 Release 2 Communications: Management Guide*, SC41-0024, provides detailed information on line protocol performance. The *Version 2 Release 2 Performance Capabilities Reference manual*, ZC41-8166, provides laboratory test results considering these parameters in various environments.

  The Chapter 9, “System Performance Tuning Tips” on page 131 section *Communication Lines considerations* of this document has general configuration parameter considerations.

10.4.1 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 screen 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 screen 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 screen functions can cause up to 2,000 bytes of data to be transferred. Although the 5294 and 5394 microcode Release 2.2 provide save/restore data compression, full screens of data may still require a significant amount of data transfer. The more frequently a save/restore screen 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 screen 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) could become excessively large. This could significantly degrade 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 then 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 screen.

Minimize Use of Break Messages during Peak Workloads: Break message support must save and restore the screen independent of the RSTDSP parameter of the currently opened display file.

The DDS Windows Support (WINDOW, WDWBORDER, etc.): Enables easier development of screen window overlays. However, the display of 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 will handle restoring screens) can be used to bypass the save and restore functions. This support requires the user program to be responsible for displaying the screen that was overlaid by windows. See the V3R1 *Application Display Programming*, SC41-3715, manual for additional details.

General Screen 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 use of Clear Line No support. You can use the Clear Line function (DDS CLRL(*NO)) when existing screen 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.

  Use of 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 EARASEINP(*ALL) 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 screen 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 lines 2 through 4 and FORMAT2 has input fields in lines 6 through 10. Outputting FORMAT 1 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 screen size (DDS DSPSIZ) is used on most workstations. The primary screen 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 screen 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 CAn.
- 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.

10.4.2 Minimizing the Amount of Workstation Specific Data Processing

There are two major areas of processing screen specific data: validity checking input data and use of AS/400 display subfile support.

10.4.2.1 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, 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, and MSGID and MSGCON.

10.4.2.2 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 then 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 (screen 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 then 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 then build the second page of the subfile and cause it to 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 would cause 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 screen 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 would be 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 could enter several input records. Even when the subfile operation ″read-next-changed″ is used, the end user may have keyed in many data records. If a large number of input records are processed by the program, a long response time could 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 would give very good response time per line item. Processing all line items before writing to the screen would give 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 use of the SFLCLR (clear subfile) and SLFINZ (initialize the subfile) keywords, rather than multiple user program output operations to clear the subfile of old records.

10.4.3 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.

The use of display file RSTDSP(*YES) and full file opens increase the number of line turnarounds. For workstation programs, the use of ″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 this waiting is called a ″short wait″ in the Performance Tools reports. If the short wait lasts longer than 2 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.

Listed below are 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. Use of this operation code eliminates a line turnaround until data is received or an error occurs.
- An output only operation, such as RPG/COBOL WRITE statement, will cause 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. Use of 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 will show 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 under various topics in this section. **This ″Write with Invite, followed by Read″ sequence does consume more CPU than the typical use of the ″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 via 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 will be displayed until one of the following is issued by the program:

- Read operation
- Write and Read (Put then Get) operation
- Write format with DDS INVITE operation

• Some applications have the need to update a screen without requiring operator input. Examples would be updates to stock status or delivery truck status screens where operator input is optional, but status changes frequent. (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 could be 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 then 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 screen. The output format must either use Put Override support or the OVERLAY keyword to ensure any data currently being keyed 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 will indicate if data has already been received. You should consider using the Version 2 DDS keyword Retain (keyboard) Lock Status (RETLCKSTS) to reduce loss of data being keyed when the screen 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 via the DTAQ parameter. The program must examine the ″read file name operation″ completed information to determine what kind of data is ready to be processed. Data queue data must be retrieved via a call to the QRCVDTAQ program. This technique minimizes unproductive examinations for updated status by the program which reduces CPU usage.

• Minimize consecutive Write-with INVITE operations when using multiple consecutive writes to the screen.

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.

10.5 Choosing Batch File Transmission Techniques

Many customers have the need to exchange large files with other systems. The AS/400 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* manual provides extensive information on many AS/400 batch file transfer facilities, but does not include all of them. 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 most of the software support available on the AS/400 for transferring files available through Version 3 Release 1. ″(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) File Transfer Subroutine support. Typically the program calling these routines is written in a separately priced high-level language.
- (OS/400) Distributed System Node Executive (DSNX) support for files with the System/390* via 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* via the NetView* Distribution Manager, another AS/400, 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 5738-CM1. This support is used for exchanging files with a System/390. Use of RJEF may require a host program on the System/390 to reconcile record length differences between the AS/400 and System/390 system 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 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.
- Personal Computer Support/400, program number 5738-PC1. Two functions are provided for exchanging file data between the AS/400 and the Personal Computer: File Transfer and Shared Folder support.
- Client Acccess/400 with appropriate client software - Original DOS, Windows, OS/2 Clients, Client Access/400 for Windows clients, Client Access/400 for OS/2 clients.

  This support also provides file transfer and network drive support, similar to the PC Support/400 support.
- TCP/IP Connectivity Utilities/400, program number 5738-TC1, with both a user program interface and a workstation operator interactive interface to File Transfer Protocol (FTP).
- OSI File Services/400, program number 5738-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 and the personal computer running AS/400 Personal Computer Support/400.

With all 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.

  Through V2R3, CPI-C interfaces are slightly more efficient than ICF file interfaces. Better performance throughput is observed with large batch file transfer applications.

  With V3R1 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 V3R1 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 V3R1.

  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.

```
 5763SS1 V3R1M0  940909                        AS/400 TRACE JOB INFORMATION                 04/03/95 15:11:53       PAGE    1
  TRACE TYPE  - *ALL     MAX STORAGE- 12000      EXIT PROGRAM-             *NONE
  RECORD COUNT- 054559   START TIME - 15:09:22   START DATE  - 04/03/95                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         410063800000000000004800000000000000000000000000000000   * Ä       ç                 *
15:10:19.878  017928  DATA      1C         41004002900080011400000000000000000000000000000000000   *   °                       *
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         F20063F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0000000000000000   *2 Ä0000000000000000         *
15:10:19.888  017933  DATA      1C         F58000000000000000000077C01784000D07404040404040404040   *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         1A0000000000000000000000000000000000000000000000000D5   *                          N *
15:10:19.894  017937  DATA      1C         1A004002900080000000000000000000000000000000000000000   *    °                      *
15:10:19.904  017938  DATA      1C         1AF0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0   * 000000000000000000000000 *
15:10:19.906  017939  DATA      75         000000000E4F01140000D1D3004040404040404040404040404040   *       |    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         000000000D3701140000D1D3004040404040404040404040404040   *            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         41C3E2E3D9C3C4404040400000001140000000000000114000000   *CSTRCD                    *
15:10:19.917  017946  DATA      1C         410063800000000000004800000000000000000000000000000000   * Ä       ç                 *
15:10:19.920  017947  DATA      1C         41004002900080011400000000000000000000000000000000000   *   °                       *
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 34. 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/400s.

  Through V2R3 (V3R0M5) AS/400 APPC is significantly faster then TCP/IP for large data transfers.

  With V3R1 batch file transfer via TCP/IP is faster than V2R3 APPC. V3R1 APPC is faster than V3R1 TCP/IP, but in some cases batch file transfers are within 1%-5% of each other.

  See 9.13.10, "General V3R1 APPC and TCP/IP Performance Expectations" on page 184 for more information on V3R1 APPC and TCP/IP performance.
- Distributed Data Management file transfer via Copy File is very 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 very 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.

10.6 APPC Programming Tips and Techniques

10.6.1 APPC Session Negotiation

Normal ″vary on″ of 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, etc.). 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%-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.

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 will require 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:

- for the non-AS/400 remote APPC programmer to 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

- determining 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 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 *prestarted jobs*

10.6.2 Maintaining APPC Conversations

AS/400 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.

10.6.3 Using PreStarted Job Support

OS/400 provides prestarted job support to speed up *incoming program start request processing.* (job initiation processing on the AS/400). A prestarted job is most useful for short transaction connections where job startup overhead is a significant portion of the total connection time. A prestart job is defined to a subsystem monitor via the Add Prestart Job Entry (ADDPJE) and Change Prestart Job Entry (CHGPJE) commands. These commands specify which programs to start and how many copies of the prestarted job to make active and maintain through the operating environment. The job′s user profile is also specified.

The prestarted jobs can be defined to start when the subsystem monitor job starts or later via the Start Prestart Jobs (STRPJ) command.

Standard OS/400 subsystem monitor support detects situations when a large number of prestarted jobs have been active followed by periods when very of few these pre-started jobs are actually doing work. When this is detected, the subsystem monitor gradually reduces the number of non-busy prestarted 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, compared to complete job startup on the AS/400.

in V3R1 some IBM applications, such as the Client Access/400 Windows and OS/2 clients, host server programs use prestarted jobs.

Refer to the *Work Management Guide* for more details on prestarted job considerations.

10.6.4 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(s) 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 wish 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, depending on the type of APPC security being used (″secured location″ yes or no), you will 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 will be 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 the *Work Management Guide* for additional information.

10.6.5 Minimize Use the ″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 only confirmation of receipt of data and no line turnaround processing is required, use the ″request confirmation-confirmed″ APPC verbs.

Note that use of ″force data,″ ″confirm,″ and 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 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.

10.7 SystemView OMEGAMON Services/400 Tips and Techniques

OMEGAMON monitoring and Automated Facilities/400 provide a set of very powerful tools. However, if many conditions are defined and/or the situations containing these conditions are examined every few seconds, excessive CPU utilization could 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, etc. A situation is either a predicate (condition and compare, sum, average, etc) 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 only 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 4 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 spool files″ collects one set of data for each output queue on the system. Redbook *Managing Operations on AS/400s with SystemView OMEGAMON Services/400*, GG24-4136, lists high to low CPU impact multiple instance attributes. Performance impact will increase if you do not specify compare values for one or more important attributes.

10.8 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 10.1, "General Application Programming Tips" on page 189 apply to RPG IV as well.

10.8.1 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. Use of 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″ support that supports familiar arithmetic operations in a single statement.

Use these new capabilities in new applications where tedious programming was required to achieve the same results, if possible.

10.8.2 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 via the IMPORT and EXPORT keywords.

Before using the new CALLB support you must understand the generally applicable ILE support constructs discussed in 10.1, "General Application Programming Tips" on page 189

10.8.3 ILE RPG Program Use of Last Record (LR)

Use of SETON LR in an RPG IV procedure running in a named activation group will leave 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.

10.8.4 ILE RPG Program Size Consideration

Given a large RPG III program, simply recreating the program into the ILE environment frequently results is longer compile times and larger disk and main storage sizes. The difference in sizes can be reduced by removing the DEBUG facility and compressing OBSERVABILITY data.

ILE program optimization can also reduce storage required. See "Optimize Programs" on page 157 for more information.

The ILE program object size can be any where between 1.5 to more 3 times the non-ILE or Original Program Model (OPM) size. even after DEBUG has been removed, OBSERVABILITY has been compressed and the ILE program fully optimized. Note that complete removal of observability is not recommended as it precludes seamless movement of the program(s) to the RISC hardware.

DEBUG information (*LIST, *SRC, *STMT) can significantly increase the object size so it should be removed when debugging has been successfully completed.

10.8.5 Run Time Working Set Size

If you determined the working set size for a V2R3 application and convert that same application to ILE compilation programs, the ILE run time working set size should be increased by approximately 256K. If you do not do this, run time in the ILE performance will degrade dramatically.

10.8.6 Compiler Working Set Size

In general the ILE compilers and translators produce more internal tables for binding and debug information than their Original Program Model counterparts. As a result the compile time working set size has increased and compile times for large programs has also increased.

The following memory sizes are recommended for ILE compilation:

- 8MB for small program compiles
- 12MB to 16MB for medium size program compiles
- 20MB for large size program compiles

10.8.7 Compiler Options

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 Commands.

For ILE RPG IV consider OPTIMIZE(*BASIC or *FULL) on the Create xxxMOD commands. *BASIC will cost less at compile time and result in small, but measurable performance improvement. In programs that are I/O intensive *FULL will show little, if any performance improvement over *BASIC.

Always start iwth OPTIMIZE(*NONE) so that performance can be compared to *BASIC or *FULL.

This will invoke 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 will also be performed.

In programs that are I/O intensive *FULL will show little, if any performance improvement over *BASIC.

Always start with OPTIMIZE(*NONE) so that performance can be compared to *BASIC or *FULL.

Decimal Data Errors: AS/400 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 will degrade performance if many decimal errors are detected by the system.

The proper resolution is to correct the fields which have invalid data.

10.8.8 Features to Use

10.8.8.1 Conditioned ″DO″ Statements

The AS/400 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 will achieve better AS/400 performance if changes to IF-THEN and DO statements are made.

10.8.8.2 Remove Invariant Variables and Subscript References

The removal of 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)
```

10.8.8.3 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 LR (Last Record) indicator *off*. With LR off, the RTRN operation (op) code allows a subprogram to remain active (all files remain open, file position retained, 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.

10.8.8.4 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, then use SEQONLY(*YES) on an Override Data Base File (OVRDBF) or Open Data Base File (OPNDBF) command before opening the file. In RPG you must define the file for sequential input or sequential output only processing. Use of either the CHAIN op code or the SETLL

(Set Lower Limits) op code will cause RPG to open the file randomly and SEQONLY(*YES) will be 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 which is being processed as sequential only.

See 10.2, “Database Tips” on page 198 for more information on sequential-only processing.

10.8.8.5 QCLSCAN, QDCXLATE and RPG Data Scanning and Translate Functions

In Version 2 RPG has enhanced data scanning capabilities via 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 programs QDCXLATE and QCLSCAN.

At this time there are no performance comparisons between the OS/400 functions and the new RPG functions. An example of the call to QCLSCAN is shown below:

```
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
```

Reference the *Control Language Programmers Guide*, SC41-8077, for more information.

If you have a very large array that is ordered, consider using the Binary Sort program in optional library QUSRTOOL. This program could significantly reduce CPU utilization.

10.8.8.6 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, etc.) will cause 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 will be performed.

10.8.8.7 For Files Opened for Update, Consider Record ″No Lock″ and ″Unlock″

Beginning in Version 2 RPG provides new ″no lock″ and ″unlock″ functions. If a file is opened for U-F (Update-full function), a ″read″ operation (CHAIN, READ, READE, etc.) 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. However, since the file was opened for update, the read will wait if another job has that record locked.

With Version 2 RPG, the unlock (UNLCK) operation now 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 via 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 then issue the UNLCK operation for the database record.

Opening a file for input-only when no records are to be updated or use of 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 index entry ″locking″ in the Version 2 *RPG/400 User′s Guide*, SC41-1348.

10.8.8.8 Declarative

When working with externally defined files the totals produced may exceed the limit for a field. This will cause incorrect reports. If you define the fields that will contain temporary values and total values with the DEFN OP-CODE then the fields will be allocated based on the current length of another field.

```
C       *LIKE  DEFN  FLDA    FLDB  + 2
```

10.8.8.9 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.

10.8.9 Features to Avoid

10.8.9.1 DEBUG Operation Code, ILE DEBUG Option

For non-ILE programs the AS/400 provides a full symbolic debugger. It is not necessary to use the RPG language debug features. The use of the DEBUG op code will cause 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 CHGPGM PGM(your program) RMVOBS(*DBGDTA).

10.8.9.2 Read Under Format (System/36 RPG)

It is recommended that if you use the design approach where you have one program simply open the workstation, write and invite input and then close the file and 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. This technique was often used by System/36 application programs.

10.8.9.3 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 (like adding 1 to a two-digit number that has the value 99). Handling the size exceptions does require additional processing even if they are ignored. The Performance Tools will summarize the number of size exceptions.

10.8.9.4 Alternate Collating Sequence

The use of an alternate collating sequence by a compiled program requires that every field involved in a comparison option (for example, COMP, IFxx, CABxx, DOxx) be translated and placed in a temporary variable prior to execution. Therefore, even if you have only one field that requires this translation from a user′s perspective, many other fields would have to be transformed too.

Examine your programs to see if the logic can be handled another way (for example, using the BITxx operation codes) to minimize the work involved.

10.8.10 System Features to Help Performance

10.8.10.1 CHGPGM OPTIMIZE(*YES)

If you created a program without the optimize option then this command will perform the optimization on the program. Use optimization on large programs that will be called frequently.

Optimization on large programs could take quite some time to complete.

10.8.10.2 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.

10.8.10.3 Record Wait (RPG II)

The S/36 allowed a single user to access a file in more than one way, and each way was allowed to update the record. On the AS/400 records are locked on a record by record basis and this is not allowed. A message will be received and you must modify the program.

10.8.10.4 Direct File Extension (RPG II)

The S/36 allowed the program to access record #400 even though the file contained only 300 records. When this happened the system would automatically extend the file. This will not occur on the AS/400 and you must program for this possibility.

10.8.10.5 Avoid Using FREE in RPG Programs

Use of FREE in RPG programs can result in disconnected file control blocks. Never use FREE with unshared (SHARE(*NO)) files.

10.8.10.6 Clearing Arrays

To clear an array in RPG, use a data structure, and the CLEAR operation code.

10.8.10.7 Use Packed Decimal Odd Length Fields

The AS/400 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 (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.

10.8.10.8 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.

10.8.10.9 Using Local Data Areas

Minimize using a Local Data Area (*LDA). ″UDS″ data areas have high overhead associated with them.

10.8.10.10 Use CHAIN Rather than SETLL Operations

Whenever possible use the CHAIN operation rather than SETLL. Set Lower Limits involves more system overhead. Note that use of either CHAIN or SETLL causes RPG to open the file for random processing, which means any database file command SEQONLY parameter is ignored.

10.9 AS/400 COBOL Tips and Techniques

Athough there are no specific ILE COBOL tips and techniques in this redbook, you should review the RPG section (10.8, "AS/400 RPG Tips and Techniques" on page 237) 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.

10.9.1 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 moves will fit 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 **know** that they will remain within the range of the table definition.

GENOPT(*OPTIMIZE): This option will request that the MI Translator spend extra time analyzing the program to attempt to do additional optimization of pointer addressing for both local code generation and global pointer assignment.

PROCESS NOTRUNC: This option significantly improves performance for programs that do extensive array processing or loop management. It is currently not documented in AS/400 COBOL publications because it differs slightly from similar support for System/390 COBOL.

The option can only be specified on the PROCESS statement that must be placed in front of the COBOL source program. This option informs the COBOL compiler that for BINARY data (COMP-4) it is *not* necessary to constrain the value (truncate) to match the PICTURE clause. This option can substantially improve the subscript processing and loop control using integer values. You *must* realize that the use of NOTRUNC has the side effect that a value outside the range of the PICTURE clause is **unpredictable and may be larger than the PICTURE specification**. Note, currently you will get a SYNTAX ERROR under SEU for this option on the PROCESS statement (ignore it and save the file and the message will go away).

```
01  A  PICTURE S999 USAGE IS BINARY VALUE 12.

would map in storage as a two byte field
    |00|0C|<- in HEX
The ANSI standard says that A will only contain the
values -999 <= A <= 999 based upon the picture clause,
but in reality it is POSSIBLE for -32768 <= A <= 32767
in a two byte binary field.
```

This option suppresses most of the PICTURE constraint checking code in the PROCEDURE DIVISION. It is intended as a performance improvement for values that stay within the range of the PICTURE specification. It is not intended (nor does it guarantee) predictable results for values outside the PICTURE range. This option will not affect values that stay within the picture specification.

BINARY variables (with the NOTRUNC option) are the best performance for data items used as subscripts or PERFORM control variables or simple integer counters. Any data with scaling (digits to right of decimal point) should not be declared as BINARY.

It is recommended that you do not use COMPUTE in calculating expressions involving binary data with NOTRUNC in effect. If the COMPUTE statement is broken down into individual steps, you may avoid forcing the compiler to generate a temporary (which is currently packed decimal) which will, in turn, force all calculations to be done in packed, which will force all binary values to be converted to packed (slower).

You should also note that COMP-4 (BINARY) data will run slower than COMP-3 (PACKED-DECIMAL) data if you do not specify PROCESS NOTRUNC because the binary data will be converted to a packed temporary to be operated upon.

PROCESS EXPINLINE: This option is not documented in standard AS/400 COBOL documentation. It causes exponentiation (**) to be generated inline. The option can only be specified on the PROCESS statement that must be placed in front of the COBOL source program. It will cause floating point to be used to calculate exponentiation (**) results. This option will produce faster code and usually improves accuracy. This is the default for releases after 1.2 of AS/400 COBOL and should not be specified

OPTION(*NOXREF): Save the overhead of creating the symbol cross-reference listing.

GENOPT(*NOUNREF): 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. The use of 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 a lot of 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.

10.9.2 Numeric Data Considerations

10.9.2.1 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 is 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, 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 be 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 will choose the appropriate data to use. Remember the CRTS36CBL command **will 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 is the additional conversions to and from packed decimal so the arithmetic operation can be performed (two-thirds fewer instructions!).

10.9.2.2 Signed versus Unsigned Arithmetic

All numeric PICTURE clauses should have a leading S. For zoned and packed decimal data, the hardware always does the arithmetic operation as a signed operation. Consider the following:

```
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.

10.9.2.3 Local Copies of Parameters

If you have parameters passed to a program, and will 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.

10.9.3 AS/400 versus S/36 Environment COBOL Differences

10.9.3.1 Invalid decimal data

One of the most common problems for people migrating to the AS/400 from the S/36 is that arithmetic on the AS/400 is performed in packed decimal (similar to the S/370). Data such as blanks and other non-numeric data when encountered in data items, will cause a MCH1202 exception when used in numeric assignment, arithmetic or compares. The S/36E Cobol compiler (CRTS36CBL) has an option, FIXDECDTA, which will cause 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.

10.9.4 Features to Use

10.9.4.1 Reference Modification

On Version 2 you should 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 now available. Since this ANSI 85 feature is only supported with the Version 2 compiler and beyond, source programs that take advantage of this cannot be compiled on back-level compiler releases.

```
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 9 times faster than moving the 10 characters one byte at a time. Greater savings can be obtained with larger strings.

10.9.4.2 Inline PERFORM

The use of inline PERFORM for looping is more efficient (approximately three times faster for the loop control code) than the use of 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
```

10.9.4.3 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 1, 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 above could be used if the valid values for I were 1, 3, or 4 and 2 would be considered in error.

10.9.4.4 EXIT PROGRAM without CANCEL

EXIT PROGRAM allows a subprogram to remain active (all files can remain open, file position retained, 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 will cause 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 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 this can happen when a program written in CL or RPG receives control from the EXIT PROGRAM.

See 10.9.4.8, “COBOL MAIN Program Prior to Primary Program/Menu” on page 250 for a programming technique to address this possibility.

10.9.4.5 Sequential Only Processing

COBOL will do automatic blocking/deblocking for a file if all of the following 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 Data Base File (OVRDBF) or Open Data Base File (OPNDBF) command preceding the COBOL OPEN file. This ensures record blocking is used to maximize throughput. See 10.2, "Database Tips" on page 198 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.

10.9.4.6 Sequential-Only Processing Beginning with Version 2 Release 2

Additional sequential blocking support is provided beginning with Version 2 Release 2. *All of the following must be 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 that it does not do prior to Version 2 Release 2:

- 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 Version 2 Release 2 *COBOL/400 User's Guide*, SC09-1383, index entry *block, description*, for more information.

10.9.4.7 READ vs 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.

10.9.4.8 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 via an EXIT PROGRAM is to be considered a STOP RUN. A STOP RUN in COBOL will cause the main program and all the programs called by that program in the entire process to be canceled (CANCEL statement) which will cause 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 will now be considered the main program of the run unit and all COBOL programs called by the driver program/menu will now be considered subprograms. These subprograms can now all terminate using EXIT PROGRAM and the files can remain open across menu transitions. (See the 10.9.4.4, “EXIT PROGRAM without CANCEL” on page 249.)

10.9.4.9 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.
```

10.9.4.10 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 will cause 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.

10.9.4.11 READ with ″NOLOCK″ for Input-Output Files

Version 2 provides an alternative to using two files (FDs). If the file is open for I-O and you read a record it will be 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 one of the following to free up the lock:

- If prior to release V2R1:

  REWRITE

- With release V2R1:

  READ . . . WITH NOLOCK

  The ″WITH NOLOCK″ clause will not lock the record but since this file was opened for I-O, the read will wait if another job has the record locked.

10.9.4.12 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. A note of caution -- SPACES are not valid zoned decimal numbers and will cause an exception when used unless they are set to a valid number.

10.9.4.13 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 not an AT END clause on the READ, control will then be 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 10.9.8, “Requesting a Formatted Symbolic Dump” on page 263 for an example of coding a declarative section.

10.9.4.14 Sorting of 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 very good idea to sort the table before you do the search. This will allow you to use 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.

10.9.5 Features to Avoid

10.9.5.1 STRING and UNSTRING

These functions are implemented as external run-time subroutines and in many cases simple MOVE statements where extra REDEFINES would do the same function much more efficiently.

With Version 2 you should consider using reference modification to work with substrings of character data.

```
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 makes it 9 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.

10.9.5.2 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. Version 1 Release 3 Extended Accept Display COBOL support (PROCESS statement EXTACCDSP parameter) gives improved performance and more control over positioning both input and output fields.

10.9.5.3 START on a Multiformat File

One should be careful 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.

10.9.5.4 Segmentation

The use of segmentation on the AS/400 will add 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.

10.9.5.5 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 5,000-line) programs. Strive for functional packaging of units of code. Very large programs take more time to compile and are harder for the compiler to optimize.

10.9.5.6 USE FOR DEBUGGING

Since the AS/400 provides a full symbolic debugger it is not necessary to use the COBOL language debug features. The use of the USE FOR DEBUGGING feature causes the compiler to generate a lot of additional code and data to accomplish the function.

10.9.5.7 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 then 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 program into one program. This technique was often used by S/36 application programs.

10.9.5.8 CANCEL

The use of the CANCEL verb causes all program data and all files associated with the program being canceled to be deactivated. This means the files will be closed and the copy of the static storage in the process will be destroyed and removed from the process. Subsequent calls to the canceled program will cause the system to re-establish the program environment within the process which is quite a bit of overhead. See ″EXIT PROGRAM without CANCEL″ for alternate ways to accomplish the function.

10.9.5.9 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 use the exponentiation operator. See the sample program for inline square root using COBOL.

10.9.5.10 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 output will indicate the frequency of exceptions on a system-wide basis.

10.9.6 Handy Things to Know

10.9.6.1 COBOL File Information Block

Even though you will get an error message at compile time, the AS/400 COBOL compiler will allow 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).
```

10.9.6.2 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.
```

10.9.6.3 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 1. This will allow you to continue processing in your program while the user keys 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.
```

10.9.6.4 Common Input-Output Handling

With the use of OPEN-FEEDBACK information and the ability to do dynamic file overrides with the AS/400, 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.
```

10.9.7 Sample COBOL Code to Do Square Root Inline

10.9.7.1 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 3O  6 43EDTWRD('    .   ')
```

10.9.7.2 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.
```

10.9.8 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

10.9.8.1 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 occured on SCREEN-FILE, File Status ="
                 SCREEN-FS "/"  SCREEN-FS-MAJOR-MINOR.
 END DECLARATIVES.
 MAIN-PROCEDURE SECTION.
    START-OF-PROGRAM.
    . . .
```

10.9.9 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.
```

10.9.10 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.
```

10.9.11 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
```

10.9.12 I-O-FEEDBACK Copy Book (IOFB)

```
*----------------------------------------------------------------*
*  I-O  FEEDBACK INFORMATION                                     *
*       ACCEPT INTO THE IOF AREA THAT MATCHES FILE TYPE          *
*----------------------------------------------------------------*
*
*  DATA BASE 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).
```

10.9.13 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.
```

10.9.14 Convert YYMMDD Date to Julian Date

```
 PROCESS NOTRUNC .
 IDENTIFICATION DIVISION.
 PROGRAM-ID. JULIAN.
*-------------------------------------------------------------*
* Convert YYMMDD date into julian date (YYDDD)                *
* Note:  CRTCBLPGM JULIAN GENOPT(*NORANGE *OPTIMIZE)          *
*-------------------------------------------------------------*
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  DAYS-SO-FAR-TABLE.
     02  FILLER PIC S9(4) BINARY VALUE   0.
     02  FILLER PIC S9(4) BINARY VALUE  31.
     02  FILLER PIC S9(4) BINARY VALUE  59.
     02  FILLER PIC S9(4) BINARY VALUE  89.
     02  FILLER PIC S9(4) BINARY VALUE 120.
     02  FILLER PIC S9(4) BINARY VALUE 150.
     02  FILLER PIC S9(4) BINARY VALUE 181.
     02  FILLER PIC S9(4) BINARY VALUE 212.
     02  FILLER PIC S9(4) BINARY VALUE 242.
     02  FILLER PIC S9(4) BINARY VALUE 272.
     02  FILLER PIC S9(4) BINARY VALUE 303.
     02  FILLER PIC S9(4) BINARY VALUE 334.
 01  DAYS-SO-FAR-REDEF REDEFINES DAYS-SO-FAR-TABLE.
     02  DAYS-SO-FAR
         PIC S9(4) BINARY OCCURS 12 TIMES.
 01  SHIFT-6 PIC S9(4) BINARY VALUE 64.
 01  HEX-ZERO PIC X VALUE LOW-VALUES.
 01  I        PIC S9(4) BINARY.
 01  DAYS PIC S9(4) BINARY.
 01  THE-YEAR PIC S9(4) BINARY.
 01  YEAR-REDEF REDEFINES THE-YEAR.
     02  FILLER PIC X.
     02  LEAP-YEAR PIC X.
 LINKAGE SECTION.
 01  YYMMDD-DATE.
     02  YY PIC S99.
     02  MM PIC S99.
     02  DD PIC S99.
 01  JULIAN-DATE.
     02  JYY PIC S99.
     02  JDDD PIC S999.
 PROCEDURE DIVISION USING YYMMDD-DATE, JULIAN-DATE.
 START-OF-PROGRAM.
     MOVE YY TO JYY, THE-YEAR.
     MOVE DD TO DAYS.
     MOVE MM TO I.
     ADD  DAYS-SO-FAR (MM) TO DAYS.
*    IF AFTER FEBRUARY SHIFT YEAR 6 BITS LEFT
*    Note: This routine does not handle the year 2000
     IF I > 2 THEN
         MULTIPLY THE-YEAR BY SHIFT-6
         IF LEAP-YEAR NOT = HEX-ZERO THEN ADD 1 TO DAYS.
     MOVE DAYS TO JDDD.
     EXIT PROGRAM.
```

10.9.15 Sample Program to Get CPU Time

```
TITLE 'CPU TIME';
;/*****************************************************/
;/* Name: CPUTIME                                     */
;/* Description:                                      */
;/*    Returns processor time used in hundredths of    */
;/*    seconds as a binary 4 value to the caller      */
;/* Linkage:                                          */
;/*    COBOL : CALL "CPUTIME" USING VAL               */
;/*            01  VAL PIC S9(9) BINARY               */
;/*    C     : void CPUTIME (int *val)                */
;/*    PASCAL: Procedure CPUTIME (var   val : INTEGER)  */
;/* Programming Language:                             */
;/*    AS/400 Assembler                               */
;/*****************************************************/
;ENTRY * (PLIST) EXT         /* Entry point           */
;DCL SPCPTR TGT@ PARM        /* address of result     */
;DCL OL PLIST (TGT@) EXT PARM MIN(1) /* Parmameter list */
;DCL SPCPTR BDST AUTO
;DCL DD VAL BIN(4)   BAS(TGT@)
;DCL DD DTA CHAR(22) BDRY(16) AUTO
;DCL DD C8  CHAR(8)  BDRY(4) AUTO
;DCL DD F8  FLT(8)   DEF(C8) POS(1)
;DCL DD S1 BIN(4)    DEF(DTA) POS(1)
;DCL DD T1C CHAR(8)  DEF(DTA) POS(13)
;DCL DD T1F FLT(8)   DEF(DTA) POS(13)
;   SETSPP BDST,DTA         /* Point to materialize area*/
;   CPYBLA C8,X"6180000000000000" /* flt value        */
;   CPYNV S1,22             /* Size of materialization  */
;   MATPRATR BDST,*,X"21"   /* Materialize exec attrs   */
;   CPYBTRLS T1C,T1C,X"000C"/* Shift right 12           */
;   MULT(S) T1F,F8          /* make it normal to micro  */
;   MULT(S) T1F,F8          /*     seconds              */
;   DIV(S) T1F,E"1.0E+4"    /* divide by 10000          */
;   CPYNV VAL,T1F           /* return the result        */
;   RTX *                   /* return to caller         */
;PEND;
```

10.10 AS/400 ILE C/400 Tips and Techniques

V3R1 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 will need to be re-translated or re-compiled in order for them to take advantage of V3R1 performance improvements.

Note that the passing of variables via a service program described under "Service Program Considerations (ILE support)" on page 192 applies to C/400 and is not included in this section.

10.10.1 ILE C/400 Function Inlining

V3R1 C/400 provides a new inline capability that allows small functions to be ″inlined″ 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.

10.10.2 ILE C/400 Exception Handling

V3R1 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.

10.10.3 ILE C/400 Heap Management

V3R1 C/400 has changed heap management and program name processing.

Heap pages are brought into main storage in 4K blocks (8 pages), instead of single pages (512 bytes). A single page is the standard AS/400 default.

In V3R1 C/400 now saves the program name after the first call to the program, so if subsequent call are made to the program in the same activation group, the overhead to get the program name will be eliminated. (ILE Call Bound program calls are still faster than this technique.)

10.10.4 ILE C/400 MUTual EXclusion (MUTEX) Instruction

V3R1 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.

10.10.5 ILE C/400 Scan Wild Card (SCANWC) Function

V3R1 C/400 provides a faster implementation for the SCANWC( ) function.

10.10.6 ILE C/400 Stream I/O Improvement

V3R1 C/400 provides a new implementation of stream I/O support available only under the CPA Toolkit (UNIX-based support) technology on preview support. March 1995 is the scheduled CPA Toolkit availability.

This implementation of stream I/O uses the Integrated File System interfaces and should provide significant performance improvement over the stream I/O implemented with OS/400 Data Management.

10.10.7 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.

10.10.8 ILE C/400 Exception Handling Recommendation

You should reduce the number of exceptions generated because exception handling is very 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.

10.10.9 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 only under 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 a whole 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 very long records. It also provides a significant performance improvement when you use Intersystem Communication Function (ICF) files.

As is true for all application programming 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 till 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.

10.10.10 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, use of pointers is more efficient than use of subscripts to access multi-dimensional arrays, as this saves a number of subscript arithmetic operations that often involve multiplications.

10.10.11 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, segment faults, and effective address overflows (EAOs). 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 16711600 bytes. 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.

You can improve your applications performance by avoiding effective address overflows (EAOs). An EAO may occur when your program references a storage location that spans across a 64K segment boundary. When an EAO occurs, it takes the system a significant amount of time to handle it. Though the AS/400 programmer does not have access to the internal implementation that may incur an EAO, the programmer can minimize occurrence of an EAO by using arrays or other storage structures that that are less than 64K in size or by not repetitively processing locations within a storage structure longer the 64K bytes that are farther than 64k bytes in displacement from each other. EAOs can cause degraded performance impact only when they occur many times within a performance measurement period, such as an interactive transaction.

The Performance Tools/400 Manager Feature reports exception counts that include EAOs. The Advisor function or tables shown in Appendix A, "Guidelines for Interpreting Performance Data" on page 323 can be used to assess CPU impact based on the CPU′s RIP factor.

10.11 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. 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 File Server I/O Processor
- 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

A complete discussion for each of these client server applications is beyond the scope of this redbook. So use the information in this redbook for performance management information, but refer to the February 1995 (or later) version of the *Performance Capabilities Reference* manual or HONE item 226NC for a more complete set of 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

10.11.1 Importance of Data and Program Processing Placement

Regardless of the client server application type the following 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 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 without expert cache and then with expert cache to evaluate any performance impact.

  See subsequent *work mangement* 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 could 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 and the client must be completed to get the best possible performance. Experience indicates 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.

10.11.2 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 the AS/400 or the File Server I/O Processor attached to the AS/400 when LAN Server/400 is being used.

- Client

  The client workstation is considered 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 client. The network drive may contain data or programs or procedures.

- Database serving

  The server contains data that the client knows and understands could 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

10.11.3 Positioning Client Access/400 and LAN Server/400

At the time of the redbook publication, LAN Server/400 supports the file server application type and many functions under administration support. Its file server performance is very competitive compared to leaders in the industry such as OS/2 LAN Server, Novell NetWare, and Windows NT servers.

Client Access/400 for the original DOS, DOS Extended, and OS/2 clients and the Windows 3.1 client provides the support for all the client server application types. V3R1 performance is greatly improved over V2R3 PC Support/400 in several areas, such as with Open Database Connectivity (ODBC) and Remote SQL. However, where top performance is critical for the file server application type, LAN Server/400 is recommended.

This topic provides recommendations comparing Client Access/400 and LAN Server/400 including:

- Database serving: performance positioning of ODBC versus Remote SQL interfaces under Client Access/400 host server support
- File serving: performance positioning of LAN Server/400 versus Client Access/400

- Administration - save/restore: performance considerations for LAN Server/400

For more detail on these topics, please refer to the *Performance Capabilities Reference* manual, February 1995 version.

Additional performance test comparisons are available to authorized IBM representatives as listed below:

- Performance Capability Reference material:

  Enter the the following statement on a VM termial:

  ```
  REQUEST V3R1 FROM FIELDSIT AT RCHVMW2 (your name
  ```

- Comparison of AS/400 LAN Server/400 and Client Access/400 file and database serving with a Hewlett Packard UNIX system and a Microsoft system with selected software

  This information is provided by the VM MKTTOOLS tools disk under package AS4CSPRF.

- Comparison of AS/400 LAN Server/400 and Client Access/400 file and database serving with a set of LAN Servers that include OS/2 LAN Server 4.0, AIX, several Novell Netware releases and other systems.

  This information is provided by the VM MKTTOOLS tools disk under package SRVPERF2.

10.11.3.1 Client Access/400 Database Serving Recommendations

Client Access/400 provides SQL interfaces that allow PC applications to directly access AS/400 database (DB2/400) information and function, such as stored procedures, rules, and triggers. The following two interfaces are provided for Windows 3.1 client applications to access AS/400 database information:

- Open Database Connectivity (ODBC, level 2 conformance), which allows client applications using the client ODBC interface to run unchanged to access AS/400 database information.
- Remote SQL, which is a set of APIs with syntax similar to SQL that were first introduced with PC Support/400 to provide remote access to the AS/400 database.

V3R1 performance of both of these interfaces has been improved significantly over the V2R3 interfaces. In addition, when comparing to the competition, the Client Access/400 ODBC driver provides very competitive performance for many database serving functions. Depending on the specific workload, many users may see **up to 30% performance improvement** as compared to V2R3, while applications using one or more of the following will see **up to two to three times performance improvement** for the following facilities:

- Multiple key index queries
- Commit levels *CS or *ALL
- Retrieving many rows
- Catalog functions
- ODBC Extended Dynamic Support (stored PREPAREed statements)

While performance has been improved for V3R1 Remote SQL and ODBC interfaces, ODBC interfaces are recommended for the following reasons:

1. ODBC is the standard industry interface for Windows applications accessing a remote database and writing to this interface will allow greater application portability.
2. ODBC will see continual performance improvement, while development support of the unique AS/400 Remote SQL interfaces has moved to ″defect only″ support.

10.11.3.2 File Serving Recommendations

Improving AS/400 file serving speed has been a top priority for V3R1 availability.

With V3R1 LAN Server/400 and the File Server I/O Processor file serving performance becomes competitive and in some cases exceeds the performance of competitive servers.

If your requirement is for high performance file serving from client workstations running DOS, Microsoft Windows, or OS/2, attached via LAN to the AS/400, then LAN Server/400 is the solution. LAN Server/400 provides PC file serving performance comparable to the best-of breed PC servers and up to *8 times* faster than V2R3 shared folder Type 2 support, while requiring dramatically less AS/400 CPU resource.

Where performance is not a key requirement in PC file serving, such as possibly in the areas of client administration or casual file serving, then file serving provided by Client Access/400 is sufficient. Client Access/400′s strength 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.

For best performance under the V3R1 Client Access/400 Windows 3.1 client, it is recommended to move data from QDLS into the Root directory under the new V3R1 Integrated File System support.

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.

10.11.3.3 Positioning Client Access/400 File Serving for DOS and Windows 3.1

It is necessary to understand where data (files and programs) reside under V3R1 to establish proper client access and performance expectations under Client Access/400.

Client Access/400 provides support for existing DOS clients (DOS Extended, and DOS). For the DOS clients, the PC data that is being shared will continue to be served from the QDLS file system (formerly known as Shared Folders). With the byte stream performance improvements provided by the new Integrated File System (IFS), the DOS clients may see equal to or faster response times with V3R1 as compared to V2R3 shared folders and will see significantly lower AS/400

CPU utilization compared to V2R3. **However, these clients will be limited to accessing data only in the QDLS file system, not the other file systems in IFS**.

The new Client Access/400 Windows 3.1 client is the recommended client to be used with Microsoft Windows 3.1 Enhanced Mode. It provides native windows client support that is easily configured/installed and runs as a Windows application. This provides tight integration with the Windows utilities such as Program Manager, File Manager, etc. It also takes full advantage of the Integrated File System as well as QDLS. From the user′s desktop, the entire AS/400 namespace will appear as a network drive. Data in each of the unique file systems (QDLS, QSYS.LIB, QOpenSys, Root, and QLANSrv) will appear as subdirectories under the Root directory. The Root file system is fully compatible with OS/2, DOS, and Windows NT file systems. In addition, 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 providing significant savings in the use of real memory.

Note that the QLANSrv file system will only be accessible if LAN Server/400 is installed.

For the Windows 3.1 client, the Root file system offers the best file serving performance. Therefore, as users migrate to using the Windows 3.1 client from the existing DOS clients, it is recommended that to get the best performance they also move their data into the new Root file system from the QDLS file system. In this case, users migrating to the Windows 3.1 client and using the Root file system can expect to see **up to a 40% response time improvement and significantly lower CPU utilization compared to V2R3 shared folder support.**

These performance improvements enable V3R1 Client Access/400 to support additional clients per CPU utilization and improved response times compared to V2R3 shared folder support.

The user must migrate V2R3 clients to the Client Access/400 for Windows 3.1 support to take advantage of access and performance improvements.

Migrating to Client Access/400

Migrating from PC Support/400 to Client Access/400 requires planning and, in some cases, use of a migration tool. The migration tool assists migrating from a Version 2 DOS PC Support/400 or Version 3 Client Access/400 Extended DOS client to a Client Access/400 for Windows 3.1 client.

The tool is a Microsoft Windows-based application and can be ordered as a PTF for the QIWSTOOL folder. The PTF order number is SF21265 for product 5763-XA1.

For more information on the migration process, refer to *Inside Client Access/400 for Windows 3.1*, GG24-4429.

Since there are a number of upgrade paths to V3R1 Client Access for V2R3 PC Support/400 clients running with Windows, Table 16 on page 283 is intended to show what performance to expect based on the path chosen. Again it is recommended to move data into the new Root file system when migrating to the Windows 3.1 client. However, this is not always possible as will be shown below in the migration options listed.

Table 16. V3R1 Client Access/400 Performance Migration Expectations	
From V2R3 PC Support/400 or V3R1 Client Access Environment	**To V3R1 Client Access/400 Environment**
V2R3 DOS Ext. (Windows), QDLS Shared Folders	Original Client-DOS Ext. (Windows), QDLS Shared Folders See Note 1.
V2R3 DOS Ext. (Windows), QDLS Shared Folders	Windows 3.1 Client, QDLS Shared Folders See Note 2.
V2R3 DOS Ext. (Windows), QDLS Shared Folders	Windows 3.1 Client, Root file system See Note 3.
V3R1 DOS Ext. (Windows), QDLS Shared Folders	Windows 3.1 Client, Root file system See Note 4.

Note:

This table shows possible ″migration options″ for data (file and programs) available under V3R1 Client Access/400 ″original client″ support and the February 1995 Client Access/400 Windows 3.1 client support (sometimes referred to as the ″Win-16 client″). At the time of publication of this redbook, insufficient performance information was available for the Client Access/400 for OS/2 clients (″premier clients″) support.

The table shows valid from and to data movement and sets performance expectations for file serving as discussed below:

1. Equal up to 40% file serving performance improvement. Significant CPU utilization reduction.
2. May see less than to equal V2R3 file serving performance. Significant CPU utilization reduction.
3. Equal up to 40% file serving performance improvement. Significant CPU utilization reduction.
4. Approximately equal file serving performance. Approximately equal CPU utilization.

In order for users to move their data easily to the Root file system, there are a number of options: AS/400 IFS move command, the Windows File Manager drag and drop support, or PC commands such as XCOPY. The new AS/400 IFS commands will provide the best performance since the function is done entirely on the AS/400.

Remember, however, to understand the job priority of the AS/400 job performing the AS/400 IFS command and what other system work is concurrently active. System impact of a high priority IFS command (job) and the amount of data being moved may require ″job scheduling.″

As always, the actual performance improvement is very dependent on overall system workloads and system configuration.

10.11.3.4 LAN Server/400 (QLANSrv) Save/Restore Performance Recommendations

LAN Server/400 integrates the administration and security of LAN-based file serving into the normal OS/400 workplace. This allows a single administrator to manage a large number of LAN services from a central location. In the area of backup/recovery, AS/400 save and restore functions can be used to save and restore data owned by LAN Server/400. This allows customers to integrate the save/restore procedures for data being served by one or more File Server I/O Processors in their normal AS/400 save/restore procedures.

At the time you install LAN Server/400 and configure a File Server I/O Processor, you define the storage spaces (network drives) that will be accessed by client workstations attached to the File Server I/O Processor. These network drives are stored in the QLANSrv file system on the AS/400. Therefore, the new Integrated File System (IFS) SAV and RST functions can be used to save and restore data on the network drives.

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, up to 20 times faster (depending on the workload). However, there are some consideration 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 File Server I/O Processor the network drive is linked to must be varied off during the save.

Regarding option 2, if you save the network drive at the file/directory level, and if a restore is required, you can selectively restore only those files needed to be restored. The File Server I/O Processor is not required to be varied off when saving/restoring at the file/directory level. As compared to saving/restoring data (folders/documents) in the QDLS file system using the SAVDLO/RSTDLO commands, saving/restoring a network drive at the file/directory level is approximately **2.5 to 3 times slower.**

Based on the above considerations, the following are recommended for saving/restoring network drives in QLANSrv:

- When configuring the network drives for an FSIOP, 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 will most likely be done at less frequent intervals than the saving of volatile data.
- For network drives with volatile data, if you need to be able to restore at the file/directory level, or you need 100% availability of the File Server I/O Processor, then you must use option 2. If you need to be able to restore at the file/directory level, but find that option 2 does not provide the required performance, you should consider the following:
  - Save using option 1 (save at the network drive level). You will have to vary off the FSIOP 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.
  - Then delete the temporary drive when complete.

Save and restore performance is very dependent on the amount of data being saved and restore.

10.12 PC Support/400 V2R3 Performance Tips

This section discusses various performance tips, including work management considerations when using PC Support/400 in releases V2R3 and V3R0M5 (Client Access/400 but V2R3 implementation).

10.12.1 PC Support/400 Performance Tips

Use Shared Folder Type 2 support for best file serving performance. Shared Folder Type 2 provides best possible file serving performance within PC Support/400 when large files are being accessed because all I/O occurs at the LIC priority.

Many short file accesses are significantly slower because the opens and closes are performed above the MI interface in the QXFSERV job.

Note that because Shared Folder Type 2 Read and Write operations are performed in the LIC, several personal computers doing large file access through Shared Folder Type 2 can ″take over the CPU.″ This is because the work is being performed at a priority higher than any user jobs and the priority cannot be controlled by the user.

Note that the PC driver code for type 2 folders takes up somewhat larger PC storage than the driver code for folder types 0 and 1.

10.12.2 PC Support/400 Work Management

Most PC Support/400 functions use programs that run in subsystem QXFPCS.

Shared Folder Type 2 Read and Write functions are performed in the LIC tasks identified in performance data as the #FSnnnnn tasks that are associated with the APPC control unit description for the personal computer. For Shared Folder Type 2 start and stop functions such as file opens and closes server job QXFSERV job runs in subsystem QXFPCS. For Type 2 support, the user can assign job priority and storage pools only to the open and close file work.

Beginning with V2R3 the Performance Monitor recorded data in the QAPMTSK database file that enable a query to join the QXFSERV job with the #FSnnnn task for each control unit. This can be used to determine CPU and disk I/O for the PC Support/400 work done for Shared Folder Type/2.

Special PC Support/400 Shared Folder Type 2 Information

Appendix G, "PC Support/400 Shared Folder Type 2 Performance Query" on page 451 contains shared folder type 2 implementation information. Sample queries of Performance Monitor database files are provided to associate OS/400 jobs with LIC tasks. This can be used for performance analysis and capacity planning shared folder type 2 work.

By manually associating the LIC task with the corresponding APPC control unit QXFSERV job into BEST/1 workloads, you can use BEST/1 for capacity planning you PC Support Shared Folder Type 2 work.

Although most of the non-workstation (Work Station Facility (WSF) and RUMBA/400) work is performed within QXFPCS, programs requested by the personal computer client enter the system through subsystem QCMN. You can change the job run priority and storage pool used by these incoming program start requests to affect PC Support/400 function performance in consideration with other system work active at the same time.

Table 17 on page 286 list the routing entries associated with PC Support/400 for V2R3 QCMN. Note that the routing entry for QPCSUPP (mode) compare position location has been corrected to Position 1. The system is shipped with the starting position of 37 (program name) which means the QPCSUPP routing entry is ignored. Manually changing the QPCSUPP position to 1 will speed up WSF startup for each attached personal computer. (V3R1 QCMN is shipped with correct starting position (01).

Routing entry for QXFINIT is not shipped with QCMN.

Table 17. V2R3 QCMN PC Support/400 Routing Entries			
Seq Number, Program - Library	**Compare Value**	**Compare Start Position**	**Function**
050 *RTGDTA - QIWS	′QVPPRINT′	37	Virtual print
100 *RTGDTA - QIWS	′QTFDWNLD′	37	File transfer
150 *RTGDTA - QIWS	′QMFRCVR′	37	Message facility receive
200 *RTGDTA - QIWS	′QMFSNDR′	37	Message facility send
210 *RTGDTA -QIWS	′QHQTRGT′	37	Data queues
220 *RTGDTA - QIWS	′QRQSRV′	37	Remote SQL
230 *RTGDTA - QIWS	′QPCSUPP′	01	Program start requests that do not satify previous comparison but using APPC mode QPCSUPP will run in QCMN subsystem. PC Support/400 WSF uses this routing entry.
240 *RTGDTA - QIWS	′QLZPSERV′	37	License management
250 *RTGDTA - QSYS	′QCNPCSUP′	37	Remote command, shared folder 0,1
260 *RTGDTA - QIWS	′QXFINIT′	37	Shared folder type 2

All PC Support/400 releases previous to V3R1 used DDM for the submit remote command and shared folder types 0 and 1 functions.

For additional information on PC Support performance, refer to *PC Support/400 Implementation and Performance*, GG24-3636.

10.13 Client Access/400 Performance Tips

This section discusses various performance tips, including work management considerations when using V3R1 Client Access/400 including the new Windows 3.1 client. Host

At time of redbook publication incomplete performance information was available for the OS/2 ″optimized client.″ For complete V3R1 performance information refer to the February 1995 or later *Performance Capabilities Reference* manual.

Use a 486 processor client whenever possible. The faster the client the better performance given the same database serving or file serving definition.

10.13.1 Client Access/400 Client Performance Tips

This topic discusses the performance tips for client configuration and client software and is separated into two major topics - database serving, and file serving.

Note that Client Access/400 provides many new and improved APIs that improve performance between the client PC and OS/400 server compared to releases previous to V3R1. Most of these improvements apply to both ODBC and remote SQL interfaces and include:

- cwbDB_Create Package() - explicit creation of an SQL package on the AS/400
- cwbDB_Prepare() - insert SQL statements into the package that are already *prepared* (fast execution when called)
- The create package and insert SQL statement create a *stored procedure* will run fast when called by another API.

- cwbDB_PrepareDescribeOpenFetch - combines preparing an SQL statement, passing it variables, opening the cursor and performing a FETCH operation
- Fetching and inserting block of records
- More efficient data stream for operations and data
- Removal of previous limits on number of open cursors and selected columns
- Supporting multiple concurrent sessions between each client and the server AS/400
- Supporting an asynchronous function that enables the client to to perform other necessary functions while the server completes a previous request.

The following topics summarize database serving and file serving performance tips for Client Access/400 V3R1 clients. Consult the February 1995 *Performance Capabilities Reference* manual for a complete discussion of performance test and tip details.

10.13.1.1 Client Access/400 Database Serving Client Tips

Under Client Access/400 the client may interface to the AS/400 database through:

- ODBC
- Remote SQL
- User-written programs to program
- Client Access/400 APIs (more function and improved performance over V2R3)

Note that in the following sections, many of the client parameter values discussed are default values. Descriptions are provided so that you can understand the implication of all valid values.

ODBC: Use ODBC interfaces for industry standard cross-system interfaces. Within the client, the application nominates which ODBC driver the Driver Manager (ODBC.DLL) is to use. For Client Access/400, you must tell ODBC.DLL to use the Client Access/400 APIs.

AS/400 ODBC API performance tips include:

- Use *parameter markers* when performing repetitive transactions. Parameter markers are variables that are passed on each SQL invocation or an already *prepared SQL statement*. A parameter marker is indicated by a "?" (question mark) in the SQL statement. where a *host variable* could appear.

  Define the client column parameter marker variables identical to host column descriptions to allow for direct mapping on the server.

- Reuse the prepared SQL statement:
  - Only *prepare* an SQL statement once and use the SQLExecute ODBC API.
  - Use SQL Package support which provides built-in reuse of prepared statements.

    It is recommended a single package be used in client applications that have a fixed set of SQL statements. An administrator should create the package and initially run the application to add the appropriate statements to the package. Configure all users of the package to not add statements, just use the package.

If each client user could add new statements at each use of the application (″ad hoc query statements″), it is recommended each client/user have their own package. Each such client/user can be configured to add statements to only their own package.

For ″user-specific″ packages, file ODBC.INI (discussed later in this topic) must be modified so each user has a different package name. Either the library name or first 7 characters of the package name can be unique.

Note, if multiple clients/users are concurrently adding SQL statements to the same package on the server, file contention will occur and performance will be degraded.

- Use stored procedures and triggers to reduce communication line flow. See index entries for these terms for additional considerations.
- Use SQLExecDirect for one way data flow (no response).

  This statement can replace the two statements - SQL Prepare and SQL Execute.
- Choose a ″fast″ client processor. 486 at 33 MHZ is the recommended minimum client processor.
- Utilize the SQL statements for blocking:
  - Use ″FOR FETCH ONLY″
  - Set the maximum frame size to at least greater than 2K for large data upload or download.

    For the Windows 3.1 client use the Global Options in Configuration.

    For Extended DOS client use the TRMS setting in CONFIG.PCS.
  - Use SQLExtendedFetch API to allow the client application to return a set of rows directly into the application from the server.
  - Use blocked inserts.
- Use the lowest satisfactory level of commitment control.
- Change pre V3R1 client applications such as CASE/4GL to use the more efficient ODBC APIs.
- Consider editing the ODBC.INI file located in the Windows subdirectory for each CA/400 ODBC client. ODBC.INI contains information necessary for accessing data through the various ODBC drivers.

  Improper setting of some information in this file may override some of the Client Access/400 ODBC API functions.
  - Set ExtendedDynamic = 1.

    ExtendedDynamic enables the ″caching″ of dynamic SQL statements on the AS/400 server. This means the statement is saved in a package after first use. Subsequent uses enable Client Access/400 ODBC to skip a signifcant amount of processing. SQL statements that can be saved include SELECT, positioned UPDATE and DELETE, INSERT with subselect, DECLARE PROCEDURE and **all other statements that contain parameter markers.**

    When ExtendedDynamic=1, the first time an application is run the ODBC driver will add a line to the ODBC.INI file for the data source that looks like the following:

```
Package<applic-name> = lib/package-name, usage, pkg full option,
                       pkg not used option
```

Once this entry has been added, it can be modified to provide the support desired.

– Set usage = 2.

  Setting usage = 2 enables the ODBC driver to use the package specified and adds statements to the package as they are run. If the package does not yet exist when a statement is added, the package is created on the server.

– Set RecordBlocking = 2.

  The RecordBlocking switch permits the client/user to control when the driver will retrieve multiple rows (block data) from the server. The preferred value of 2 enables blocking for for all SQL I/O SELECT statements unless the statement specifies ″FOR UPDATE OF″ clause.

– Set BlockSizeKB = 32 or larger.

  BlockSizeKB controls the number of rows that may be fetched in a block per each transmission between the client and the server AS/400.

  The value represents the client buffer size in multiples of 1024 byte (1KB) blocks. This size is divided by the row length to determine the number of rows that can be sent in a single send or receive. A large buffer size speeds up queries that exchange large amounts of data.

  The default value of 32 (32KB) will perform well under most environments. If the client has sufficiently large memory a value may improve performance.

– Set LazyClose = 1.

  The LazyClose switch enables the client/user to control the way SQLClose commands are processed by the Client Access/400 ODBC driver. The default value of 1 enables the delaying of the SQLClose command to the AS/400 until the next ODBC request is sent.

  This is a performance option that minimizes line turn arounds as the SQLClose would flow separately if LazyClose = 0.

– Set ForceTranslation = 0.

  The ForceTranslation switch allows users to control the Client Access/400 ODBC Driver processing with columns found on the AS/400 with an explicit CCSID value of 65535. The default setting of 0 avoids translation of the data to ASCII.

– Set Library View = 0.

  The LibraryView switch allows users to control the Client Access/400 ODBC Driver processing of certain catalog requests that ask for all AS/400 SQL tables on the system. The default value of 0 will cause catalog requests to use only the libraries specified in the default library list to be processed.

  A value of 1 will cause all libraries on the system to be processed by the catalog request and could result in significant performance degradation on a server with a large number of files and libraries to be processed.

10.13.1.2 Client Access/400 File Serving Tips

While support under LAN Server/400 provides the fastest file serving performance on the AS/400, there will be many situations where customers find adequate file serving performance under Client Access/400 support. The tips included in this topic will assist in maximizing Client Access/400 file serving performance.

Tips for All Clients

- Consider the following client cache size considerations:
  - Use small cache (256KB - 1MB) for accessing data sequentially in small amounts.
  - Use medium to large cache (500KB or larger) for accessing data randomly.
  - Use medium to large cache (500KB or larger) for accessing data both sequentially and randomly.
  - At some point an increase in cache size no longer improves performance.

    You should experiment to find the optimum size. Note that a large cache size reduces communication flow with the AS/400 and reduces AS/400 CPU utilization in busy environments.
  - If you typically run your applications once a day and use the data associated with these applications once a day, the cache size recommended is the maximum cache size determined for those applications.
  - If you typically run the applications and associated data multiple times through the day, the cache size recommended is the sum of the cache sizes determined for those applications.
- Consider LAN frame size and LANMAXOUT values.

  Use the largest frame size possible. Remember the smallest frame size specified on either the host server or the client is the frame size used.

  Set the AS/400 APPC controller description LANMAXOUT = 7, for PS/2 models 50, 60, 70, and 80 attached via LAN.

  See 9.13, "Communication Performance Considerations" on page 168 for more information on these parameters.
- Use a 486 processor client.

  Minimal performance improvement for V3R1 will be realized with a 386 processor client.

Tips for DOS Clients

- Consider the following client cache size considerations:
  - Client Access/400 provides the tool GETSTAT that can assist tuning client cache size.

    Download GETSTAT from AS/400 folder QIWSTOOL by running program IWSTOOL in QIWSTOOL. GETSTAT can show memory used by your IFS File Serving programs and buffers and indicates how effectively the cache is working at avoiding data transfer with the AS/400.

    GETSTAT only works with the DOS Extended client.

- When using DOS Extenders (XMS), use a minimum of 128KB of cache.
- The value specified on MCAC, MCAE, and MCAX identifiers are used to create both the IFS File Serving cache and a table to track contents of the cache.
- When using DOS Extenders, space for the cache as specified on the MCAX parameter is allocated from extended memory. Using XMS does not permit cache and associated cache table allocation in conventional memory.

- If using DOS 5.0 and have ″dos=high,umb″ statement in CONFIG.SYS file, it is possible for IFS File Serving code to use only 96 bytes of conventional memory.
- Communications buffer size affects IFS file serving performance.

  The size is specified with the CBSX identifier in the CONFIG.PCS file. This buffer size can be changed only if it is located in conventional memory with the exception for SDLC and ASYNC protocols that support placement in extended memory.

  If there is sufficient client memory increase the communications buffer to improve performance. If you encounter memory usage problems, change the communications buffer back to its default size of 8KB.

Tips for OS/2 Clients: In heavy file I/O environments where high file serving performance is critical consider using LAN Server/400, OS/2 LAN Server, PC LAN program, or Netware for SAA rather than Client Access/400.

Consider Client Access/400 only when top performance is not critical or the customer requires the Client Access/400 functions of file transfer, host integration, 5250 emulation and remote system access.

Client Access/400 can reside simultaneously with one of the above file servers in the same client workstation. This enables best file server performance and key Client Access/400 functions to be used as needed. For example, use one high speed file server for program loading while using Client Access/400 for data storage or exchange.

File Serving Application Tips: In almost all cases the PC hard disk delivers best performance. However, there are application techniques that can minimize the difference between the hard disk and a ″remote server disk.″

- Place PC programs on the hard disk.
- Copy files to a PC RAM disk or hard disk or use files already on the PC.

  Copy the changed data to the AS/400 when all file processing has completed.
- The PC application should be designed to:
  - Use appropriate create or open operation instead of doing a search.
  - Design the application to open, process and then close a file.
  - Use ″write verification″ only when absolutely necessary.
  - When reading a large number of records, read sequentially instead of randomly wherever possible.

    This maximizes blocking of data.

  - Read small files into memory once for repetitive accesses, rather than reading from disk or the server AS/400 for every access.

- When backing up files consider:

  - Backing up large numbers of files or folders when system activity is low.

  - Use the OS/400 CHGJOB command to lower run time priority of the backup job.

  These tips reduce the AS/400 impact when backing up files to a folder.

- For new clients, use file systems Root and QOpenSys for best IFS file serving performance.

- Avoid frequent open/delete functions for the same Root or QOpenSys directory.

  Delete causes the name cache to be invalidated which degrades performance. Keep directories that have frequent deletions (such as temporary files) separate from directories containing files rarely deleted.

- Avoid administration functions during peak IFS activity.

  Administration degrades cache efficiency,

- Use Root or QOpenSys hard links wherever possible.

  However, at some point a large number of hard links to the same file can degrade performance. Best results are when a few hard links exist for the most frequently used files.

- For optimized OS/2 clients, use the ″current working directory.″

  By assigning a drive to the working directory or using the cd (Change Directory) command you define the current working directory. This reduces the number of directory look ups.

- Minimize use of mkdir (Make directory) and rmvdir (Remove Directory) when using the Root or QOpenSYS file systems.

- Minimize create and delete when using the QDLS file system

10.13.2 Client Access/400 Work Management

V3R1 Client Access/400 server implementation, QCMN and QSERVER subsystem configuration parameters, and user documentation in *OS/400 Server Concepts and Administration*, SC41-3740, enable the following performance management capabilities:

- Control of Client Access/400 program/job run priority and storage pool assignment based on the customer′s priority for each function.

  User documentation identifies the program(s) used for each function.

- Measurement of CPU and disk resource utilization without the difficulty of identifying LIC tasks that perform significant function.

  In contrast to V2R3 PC Support/400 shared folder type 2 read/write functions actually performed by #FSnnnn LIC tasks, most of the the Client Access/400 work for each function is included in the appropriate OS/400 job.

- BEST/1 capacity planning of Client Access/400 functions without the effort of complex queries to associate multiple jobs and LIC tasks.

Since most of the work is associated with OS/400 jobs, these jobs can be associated into BEST/1 workloads according to the functions to be modeled together, such as only file serving or only database serving.

OS/400 provides server programs and jobs that support pre-V3R1 clients, V3R1 ″Original Clients″ (Client Access/400 for DOS, for DOS with Extended Memory, and for OS/2), new V3R1 Client Access/400 for Windows 3.1 clients, and new V3R1 Client Access/400 Optimized for OS/2.

A complete discussion of Client Access/400 work management is beyond the scope of this redbook. You must review the *OS/400 Server Concepts and Administration*, SC41-3740, for a more complete understanding of the servers. However, the following is provided to permit controlling Client Access/400 function job priority, storage pool assignment, and grouping of jobs into workloads for performance measurement and capacity planning:

- Summary of OS/400 host servers
- Subsystem Used
- For each subsystem a table showing routing entry compare values for each Client Access/400 function.
- List of the class descriptions used by Client Access/400.
- Discussion of Client Access/400 prestart job support.
- Consideration of more flexible user exit programs through use of the V3R1 OS/400 ″registration facility.″

10.13.2.1 Summary of OS/400 Client Access/400 Host Servers

PC Support/400 servers maintained in V3R1 include:

- File transfer
- Remote SQL
- Data queues
- Virtual print
- Message function
- License management
- Submit remote command

Shared folder type 0, 1, 2 support is still provided for previous release clients, but for V3R1 clients the support is implemented completely different via new database and file server programs. Also a new Network print server has been implemented.

The new for V3R1 servers include:

- Database server

  This server performs as an ODBC server and a Remote SQL server more efficiently than V2R3/V3R0M5 implementations. It also supports retrieval of database and SQL catalog information. These functions run in new V3R1 subsystem QSERVER.

- File server

This server replaces all previous release shared folder support. There is no ″shared folder″ terminology for V3R1 Windows 3.1 and optimized OS/2 clients.

Clients can store and access files and programs located on the AS/400 through this server. This server can access the Integrated File System Root, QOpenSYS, QSYS.LIB, and QDLS file systems. (LAN Server/400 is required to access the QLANSrv file system.) Client Access/400 clients use their own interfaces to interact with these file systems.

For all client types there is one file server job per client. Transaction programs are: QWPFSTP0 (used for original clients), QPWFSTP1 (used for Client Access/400 optimized OS/2 clients), QPWFSTP2 (used for Client Access/400 Windows 3.1 clients).

The file server jobs run in subsystem QSERVER.

- Data queue server

  This server is more efficient than previous release data queue support and runs in subsystem QCMN by default.

- Network print server

  This server provides a superset of the previous release virtual print functions including print data, process print messages, manage spool file entires and attributes, manage writer and output queue, etc. The server runs in subsystem QCMN by default.

- Remote command/distributed program call server

  This server provided previous release submit remote command support and additional ″remote procedure call″ support. The results of the command or the program on the AS/400 can be returned to the PC application.

  This server runs in subsystem QCMN by default.

  Note that there is no server job for the original client shared folder type 0 and type 1 and remote command support. They use target DDM support and appear as DDM work in the Performance Tools/400 reports.

- APPC password security management server

  This is a new V3R1 function. The server provides improved security for client users by not allowing client users with expired passwords to access the system.

  This server is invoked at the initial client connection to the AS/400 and runs in subsystem QCMN.

- Central server for:

  – License management

    The initial request from a client validates a license for each Client Access/400 user. The server job remains active until the client disconnects. At disconnection, the license is available for use by another client.

  – Client (″who is attached″) management

    When a new client attaches to this server a record is updated in the client management database. When connected the database record for the client is marked active. After disconnection, the client status is changed to inactive.

- Character conversion mapping

  Conversion maps are provided for clients to convert between EBCDIC and ASCII character representation. The client may also request the correct CCSID character code conversion map from the server by giving the correct source and target CCSID.

This server runs in QCMN subsystem.

10.13.2.2 OS/400 Client Access/400 Subsystem and Job Information

As described above Client Access/400 jobs run in either subsystem QCMN or QSERVER (new for V3R1). The old PC Support subsystem QXFPCS is no longer available.

Table 18 on page 296 shows the V3R1 QCMN subsystem routing entry compare values for V3R1 Client Access/400 host server programs. Most, but not all of the routing entries are shipped with the V3R1 release. The function description will note when a subsystem routing entry is **not shipped with the V3R1 software release.**

Table 18. V3R1 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, etc.
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)
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 will 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 will 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 will initiate a program in QCMN.
310 *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 could be many jobs from a wide variety of applications, such as Client Access/400, ADSM/400, Pass-through, Distributed Data, customer applications, etc., you should consider adding storage pool(s) 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 above 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 19 lists the routing entry compare values for QSERVER subsystem which provides all Client Access/400 file server and database server jobs.

Table 19. V3R1 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)
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

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.

10.13.2.3 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 wishes within the limits of the exit points supported by Client Access/400.

For V3R1 Client Access/400 provides more exit points than previous releases and implements user access to these exit points through the new for V3R1 OS/400 Registration Facility.

From a performance viewpoint the V3R1 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.

10.14 LAN Server/400 Performance Tips

LAN Server/400 and the File Server I/O Processor (FSIOP) offer file serving performance competitive to OS/2 LAN Server, AIX LAN Server and equivalent non-IBM servers. Performance tips are included in this topic. However, you should review the following documentation for complete coverage of LAN Server/400 performance test results and considerations:

- February 1995 *Performance Capabilities Reference* manual, ZC41-8166
- *LAN Server/400: A Guide to Using the AS/400 as a File Server*, GG24-4378
- VM Marketing Tools Packages:
  - Comparison of AS/400 LAN Server/400 and Client Access/400 file and database serving with a Hewlett Packard UNIX system and a Microsoft system with selected software

    This information is provided by the VM MKTTOOLS tools disk under package AS4CSPRF.
  - Comparison of AS/400 LAN Server/400 and Client Access/400 file and database serving with a set of LAN Servers that include OS/2 LAN Server 4.0, AIX, several Novell Netware releases and other systems.

    This information is provided by the VM MKTTOOLS tools disk under package SRVPERF2.

This topic provides a description of performance-related FSIOP considerations and sample queries of the FSIOP and LAN Server/400 performance monitor data collected in file QAPMIOPD.

The queries can be used to determine read and write performance metrics. The query results must be used in conjunction with observed performance from the client user's perspective.

File Server I/O Processor Performance Data

FSIOP token ring or Ethernet LAN adapter line utilization and error rates are recorded and reported just as for other LAN adapters - System Report and Resource Report. The Performance Tools/400 Advisor function and BEST/1 treat the FSIOP LANs just as any other LAN hardware.

The V3R1 Performance Tools/400 list FSIOP utilization in the communications IOP section of the System and Resource Reports. The reported FSIOP utilization data is for the FSIOP internal "pipe task IOP," that supports data read/write between the AS/400 disks and FSIOP storage. This IOP performance data is collected in Performance Monitor database file QAPMIOPD and is indicated as a '1' in the XIDTYP field.

The Performance Monitor also collects other FSIOP data into database file QAPMIOPD. This includes HPFS386 (High Performance File System) data (XIDTYP=3) and modified OS/2 CPU data (XIDTYP=2). However, Performance Tools/400 does not include this collected data in any reports.

You must use queries to obtain FSIOP modified OS/2 CPU utilization and cache hit rates which are key to assessing performance capacities. FSIOP CPU utilization and cache hit rate guidelines are included in this section.

10.14.1 LAN Server/400 and FSIOP Performance Considerations

The following information is based on Rochester lab conducted tests and IBM Germany tests documented in the Marketing Tools (MKTOOLS) packages described earlier in this section.

- FSIOP and Server Model Systems

  Since the FSIOP attaches to all AS/400 models, there is no requirement that the FSIOP be attached to a server model. Making a choice between a traditional model and a server model should be made on the basis of what kinds of applications will actually run on the AS/400 CPU.

  The FSIOP implementation enables concurrent file serving via the FSIOP cache and standard LAN access to Client Access/400 from the same client.

- Number of Clients per FSIOP

  The primary determiners of FSIOP file serving performance are the amount of memory available for cache memory after all configuration has been completed and the rate of requests received from attached clients. More clients may be attached when doing casual file serving versus heavy file serving. An example of heavy file serving would be multimedia continuous medium quality video from several clients at the same time.

  Use the following table as a starting point for determining an estimate of the number of attached clients supported by a single FSIOP.

Table 20 (Page 1 of 2). Guideline for Attached Clients

FSIOP Memory Size	Range of Concurrent Users
16MB	1 - 20

Table 20 (Page 2 of 2). Guideline for Attached Clients	
FSIOP Memory Size	**Range of Concurrent Users**
32MB	20 - 50
48MB	50 - 100
64MB	100 - 250

- Amount of FSIOP Memory Available for Cache

  Key for LAN Server/400 performance is the amount of cache memory available for the HPFS (High Performance File System - modified from OS/2) to store file and program data. The following table shows the available cache memory based on FSIOP memory size and the number of LAN attachments (1 or 2) attached to the FSIOP.

Table 21. FSIOP Cache Size Available for HPSF

FSIOP Memory Size	Available Cache for HPFS data
16MB	4.8MB (1 port), 2.8MB (2 ports)
32MB	16.3MB (1 port), 13.8MB (2 ports)
48MB	28.6MB (1 port), 25.6MB (2 ports)
64MB	43.6MB (1 port), 40.6MB (2 ports)

Note:

- Redbook *LAN Server/400: A Guide to Using the AS/400 as a File Server* contains more information on this subject.
- The LAN Server/400 DSPNWSSTG (Display Network Server Storage) command displays active storage assignments within the FSIOP.
- The OS/400 WRKHDWPRD (Work with Hardware Products) command shows the memory size of the FSIOP and what LANs are attached. Both WRKHDWPRD and WRKHDWRSC will show the FSIOP as 6506 with a single LAN as 6510, a two LAN feature as 6520.

- AS/400 Machine Pool Storage for an FSIOP

  For a single LAN FSIOP you should allocate 2700K bytes to the machine pool. For a two LAN FSIOP, you should allocate 4200K bytes of storage to the machined pool.

- Cache Hit Ratio of 90% or Higher Guideline

  Internal lab performance tests indicate top performance is associated with a cache hit ratio of 90% or higher. In general the FSIOP CPU utilization will not be a bottleneck. Either a cache hit ratio below 90% or utilization of the LAN itself will be the critical bottlenecks in a heavy file serving environment.

  Lab tests using the BAPCO client server benchmark of applications show the 64MB FSIOP can deliver equal to or better file serving than OS/2 LAN Server and OEM severs as documented in the *Performance Capabilities Reference*. BAPCO is made up of a repetitive series of transactions for applications such as Freelance Graphics, Harvard Graphics, Word for Windows, Word Perfect, Paradox, Excel, Lotus 1-2-3, and a mathematical spread sheet (product name may not be published).

  If you can configure a LAN configuration and workload with an OS/2 LAN server, you can use the ″CACHE386 /stats″ command that shows cache statistics for reads and writes as a base metric to compare to FSIOP cache read and write statistics. The OS/400 Performance Monitor records FSIOP CPU, disk, and HPFS statistics in file QAPMIOPD.

  You must create your own query to provide the FSIOP cache statistics. 10.14.2, “FSIOP Performance Monitor Data Queries” on page 303 contains examples of a query definition for FSIOP HPFSCache data (field XIDTYP=3) and output of the query.

- FSIOP CPU Utilization 80% or Less Guideline

  Internal lab performance tests indicate top performance is achieved when FSIOP cache hit percentages are above 90% and the FSIOP CPU utilization is 80% or less.

  You must create your own query to provide the FSIOP CPU statistics. 10.14.2, "FSIOP Performance Monitor Data Queries" on page 303 contains examples of a query definition for FSIOP CPU data (field XIDTYP=2) and output of the query.

- AS/400 CPU Utilization and Disk Busy

  As the cache hit ratio decreases below 90%, the AS/400 CPU utilization and disk busy percentage will increase as the FSIOP must retrieve data from the AS/400.

  In situations where cache hit ratio is in the 90% range, CPU utilization should be in the 3-5% or less range.

  Note that the FSIOP contains a ″pipe task CPU″ just for accessing AS/400 storage management with a special interface to AS/400 disks just for the FSIOP.

- FSIOP LIC Tasks and OS/400 Job Name

  There is an AS/400 monitor job for each active FSIOP. The job has the name of the active Network Server Description and will run in subsystem QSYSWRK.

  The following LIC tasks will be active for each FSIOP. These are the tasks that account for the CPU that may be consumed as a result of FSIOP disk accesses:

  1. #Onnnn - DASD I/O Server task to map HPFS space to AS/400 disk space

     There may be up to four tasks per FSIOP.

  2. SMnnnn - Storage Management task to perform physical disk I/O

     There may be up to two tasks per FSIOP. The nnnn values for related #O and #SM tasks are in sequential order. For example #O0014, #O0015, #O0016, #O0017, #SM0014, and #SM0015 would be related to the same FSIOP

  3. ROUTxx - IPCF (IOP Interprocess Communication Facility)

     This is standard LIC support to handle communications across the SPD bus and the attached FSIOP.

  4. FSIOP to FSIOP Interconnection Tasks - #FPNTASK, #FPNDRIVR

     These LIC tasks will be active if a client attached to one FSIOP is accessing data stored in a second FSIOP on the same AS/400.

- LAN Server/400 NetBIOS Parameters

  LAN Server/400 uses NetBIOS communications over the LAN. The CRTNTBD (Create NetBIOS Description) command contains various NetBIOS LAN protocol timers, retries, buffer size and transmit/receive values. Take the defaults unless you are a NetBIOS expert.

- LAN Server/400 Storage Link Parameters

LAN Server/400 provides some HPFS buffering parameters that affect performance. The defaults on the ADDNWSSTGL (Add Server Storage Link) command are recommended. Do not change them unless you are an OS/2 LAN Server performance expert.

These storage link parameters are described below.

– Lazy Writes (LZYWRT)

  Lazy Writes specifies whether this network server performs lazy writes of modified cache buffers to disk for this storage space. Using lazy writes enhances system performance by allowing cache blocks to be updated multiple times before they are written to disk. Lazy writes may be disabled if data integrity in the event of system failure is a major concern. Enter *YES perform lazy writes on this server. Cache blocks are written to DASD within the specified maximum age and buffer idle time values. Entering *NO forces the cached blocks to be written to DASD immediately after being updated.

```
Default value    *YES
Range            *NO or *YES
```

– Maximum Buffer Age (MAXAGE)

  Maximum Buffer Age 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 DASD when a lazy write cache policy is used.

```
Default value    5000 milliseconds
Range            1-32767 milliseconds
```

– Buffer Idle (BUFIDLE)

  Buffer idle is the cache buffer idle time in milliseconds for this storage space. This is the minimum time data remains in the cache before being written to DASD when a lazy write cache policy is used.

```
Default value    500 milliseconds
Range            1-32767 milliseconds
```

We recommend using the default NetBIOS and LAN Adapter parameters although these can be changed. These parameters should only be changed should you require tuning LAN Server/400 according to special requirements on your network:

– Bridges and Routers

– Wide Area Links

– Requester's Adapter Configuration

– Requester's Configuration

• Downloading LAN Requester Support from Client Access/400

  LAN Server/400 software includes the required LAN Requester programs and files required for the attached client. The redbook *LAN Server/400: A Guide* *to Using the AS/400 as a File Server*, GG24-4378, provides instructions on how to use Client Access/400 to download the LAN Requester support. This is done by copying the LAN Requester support from the QLANSrv file system to the QDLS file system where it can be accessed via shared folders.

The copy to QDLS process includes using the Remote Command support:

```
RMTCMD /I <d:\path>LS4DISKS.TXT /Q
```

The LS4DISKS.TXT file has many Make Directory, Change Directory, and Copy commands. The command functions and amount of data to be copied are quite extensive. If you issue the RMTCMD from an interactive session at priority 20 or higher, overall system performance is degraded during this process.

We recommend scheduling this process so as not to interfere with the normal production environment. See ″Downloading LAN Requester Code to Diskettes″ in *LAN Server/400: A Guide to Using the AS/400 as a File Server*, for more information.

10.14.2 FSIOP Performance Monitor Data Queries

Figure 35 on page 304 shows the query definition when selecting ″HPFS data″ from Performance Monitor database file QAPMIOPD. Figure 36 on page 306 shows the query output for a specific collection of FSIOP ″HPFS data″ from Performance Monitor database file QAPMIOPD. Refer to the *Work Management Guide* for field definitions. Field XIDTYP containing a ″3″ identifies HPFS data.

```
5763QU1  V3R1M0  940909           IBM Query/400           RCHASLA6  3/13/95   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                     3/13/95   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
```

Figure 35 (Part 1 of 2). QAPMIOPD File Query Definition - CACHE

```
  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
  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                      3/13/95   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                      3/13/95   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 35 (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 . . . . . . . . 03/13/95
                                                    TIME . . . . . . . . 16:53:37
                                                               HPFS386 Statistics- Cook 02
```

03/13/95 16:53:37 — LAN Server/400: HPFS Cache Statistics and File Open/Close — PAGE 1

Int #	Int Sec	IOP Bus	Data Type	Total # READS	CACHE HIT % READS	Total # Writes	CACHE HIT % WRITE	Files Opened	Files Closed	Read Reqs frm CACHE	Read Reqs from DISK	Write reqs from DISK	Write Reqs LAZY 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 36. QAPMIOPD File Query Report - CACHE

Figure 37 on page 307 shows the query definition when selecting ″CPU data″ from Performance Monitor database file QAPMIOPD. Figure 38 on page 308 shows the query output for a specific collection of FSIOP ″CPU data″ from Performance Monitor database file QAPMIOPD. Refer to the *Work Management Guide* for field definitions. Field XIDTYP containing a ″2″ identifies FSIOP CPU data.

```
5763QU1  V3R1M0  940909           IBM Query/400           RCHASLA6  3/22/95   13:59:07        Page    1
  Query . . . . . . . . . . . . . . . . . CPU486COOK
    Library . . . . . . . . . . . . . . . DJOHNSON
  Query text  . . . . . . . . . . . . . . FSIOP 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                     3/22/95   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 37 (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                      3/22/95   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 - FSIOP CPU

Page headings and footings
  Print standard page heading . . . . . . Yes
    Page heading
      8 Interconnected LAN Server/400 Clients - FSIOP CPU
    Page footing
```

Figure 37 (Part 2 of 2). QAPMIOPD File Query Definition - CPU

```
                                                  8 Users Interconnected - FSIOP CPU
                                          QUERY NAME . . . . . CPU486COOK
                                          LIBRARY NAME . . . . DJOHNSON
                                          FILE         LIBRARY      MEMBER       FORMAT
                                          QAPMIOPD     FSCITY2      BAP8WS64MB   QAPMIOXR
                                          DATE . . . . . . . . 03/22/95
                                          TIME . . . . . . . . 13:59:07
                                                FSIOP 486 CPU Utilizationt Statistics


03/22/95  13:59:07  8 Interconnected LAN Server/400 Clients - FSIOP 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 38. QAPMIOPD File Query Report - CPU

10.15 ADSTAR Distributed Storage Manager/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. Clients running the appropriate ADSM client support include the following operating systems:

- Apple** Macintosh** OS
- DEC ULTRIX**
- DOS**

- Hewlett Packard** HP-UX
- IBM AIX/6000
- IBM OS/2
- Microsoft* Windows*
- Novell* NetWare**
- SCO** UNIX 386/SCO Open Desktop**
- Sun Microsystems** SunOS**/Solaris**

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. Redbook *Setting Up and Implementing ADSTAR Distributed Storage Storage Manager/400*, GG24-4460, and the February 1995 *Performance Capabilities Reference* manual 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″ will be faster than going directly between the client and a tape device.
- The initial backup of a client is a full backup and will take 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 will be significantly faster than an entire 25MB directory made of 100 files within several subdirectories.
- When implementing a client backup approach, start out running 5 or less clients within a time period while collecting Performance Monitor data. Use the information contained in 10.15.2, “ADSM/400 Work Management” on page 312 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.
- 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 decrease total run time for all backups.
- APPC protocol is faster than TCP/IP.

  V2R3 TCP/IP support is not recommended as its large data transfer performance is significantly slower than V2R3 APPC and V3R1 TCP/IP support.
- The faster the client processor the better throughput performance.

  For example in one IBM laboratory test with an AS/400 F70 running V3 and backing up a 25MB file delivered the sample throughput rates.

Table 22. V3R1 ADSM/400: Sample 25MB File Throughput

Client Hardware-Software	Throughput Rate (KB/Sec)
PS/2-80-MS Windows	057
PS/2-80-NSDOS	164
PS/2-80-TCP/IP	157
PS/2-80-APPC	186
PS/2-95-TCP/IP	430
PS/2-95-APPC	443
AIX/6000 TCP/IP	492
AIX/6000 APPC	494

  Note this table shows maximum ADSM/400 throughput with a single large file and no other AS/400 system activity. See a following item for a more typical scenario when 25MB of data are backed up, but the data is composed of several smaller files and sub-directories.
- Use the following table to set expectations for a more typical directory, sub-directory and multiple small files backup from a single client.

  The throughput rates are based on laboratory tests with either V3R1 TCP/IP or V3R1 APPC protocols and an OS/2 PS/2 Model 95 client.

Table 23. V3R1 ADSM/400: 25MB ″Tree″ Throughput Expectations

Requirement	50KB/sec	100KB/sec	150KB/sec	200KB/sec
AS/400 Minimum System Required	F45	E60/F50	D80	E90

Note:

- In general you may consider the rates shown as ″maximum aggregate throughput″ rates.

  For example, if two clients attached to the same E60/F50 were being backed up at the same time with the same tree structure you could expect each client to achieve a 50KB/sec rate (aggregate = 100KB/sec).

- The directory and file ″tree″ used to set expectations was defined as:
  - 401 files (file sizes range from 1 byte to 12MB)
  - Files reside 2 to 5 directories from the root.
  - 44 directories and subdirectories

As you can see, the ″tree″ throughput rates are significantly slower than the single large file rate previously listed.

- Other ″file transfer applications,″ such as V2R3 PC Support/400 and Client Access/400 can deliver significantly faster throughput than ADSM on an individual file or directory test.

  These other applications provide no history or automated scheduling of the backups. It is this scheduling and history and the capability to provide backup and restore for many IBM and non-IBM clients that is the high value of ADSM/400.

  However, as previously discussed it is very important to consider all the performance tips listed and do some 2-5 client backup scenarios. Analyze the Performance Tools/400 reports before committing to a backup environment involving 20 or more clients.

10.15.1 ADSM/400 AS/400 APPC and TCP/IP Configuration

The *Performance Capabilities Reference* manual contains many performance test results based on the various AS/400 and client parameter values. The following are presented as good ″starting values″ for ″best″ ADSM/400 performance.

10.15.1.1 ADSM/400 SNA and TCP/IP Parameters

The following 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

10.15.1.2 ADSM/400 via 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

10.15.2 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 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 39 on page 313 represents the QADSM jobs and QCMN jobs that would be 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 transfer rate taken every few seconds. In general, the rate(s) shown will indicate 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 screen.

QADSM Subsytem 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
QANRSERV	WBL	BCH	PGM-QANRSERV	TIMW
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	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	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

QCMN Subsytem

Job Name	User	Number	Type	Function
ADSMCLIENT	QADSM	127735	BEVOKE	PGM-not avail

Figure 39. ADSM/400 Subsytem 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 sbusystem job names are identical - no definition of their specific function. You may collect Performance Monitor data during backup, restore, etc. 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 only 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 will run 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 recommend routing entry sequence for mode QADSM is as follows:

```
Seq Nbr    Program    Compare Value          Start Positon

 290      *RTGDTA     'QPCSUPP'                        1
 295      *RTGDTA     'QCASERVR'                       1
 299      *RTGDTA     '#INTER'                         1
 300      *RTGDTA     'QADSM'   <-- Class QBATCH      1
 310      *RTGDTA     'PGMEVOKE'                      29
```

10.16 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 screen 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 at the same time.
- Print serving

  Any AS/400 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 retrived from the NetWare server to the AS/400. 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 via 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 higher. Performance information in this section is limited to OS/400 work management information using SNA APPC as described in the next topic.

10.16.1 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.

10.16.1.1 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 via a Submit Job command to any subsystem.

10.16.1.2 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 NetWare print queue is associated with an AS/400 spool output queue.

On the AS/400 there are two APPC sessions for print serving and these jobs run in subsystem QLANRES.

10.16.1.3 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 Subsytem

   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 Subsytem

   Job Name     User         Number   Type       LANRES
                                                 Function
    HL01Q       QSECOFR      129958     B      Host->Lan Prt
                                  3
```

Figure 40. LANRES/400 Subsystem Jobs

Figure 40 shows the APPC send/receive pair jobs in 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 - display lock to verify the APPC device description for an attached server 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.

10.17 ImagePlus WAF/400 Performance Tips

At the time of publication there was no performance information available on V3R1 support of the ImagePlus Workfolder Application Facility/400. The following tips apply to releases prior to V3R1 and **may apply to V3R1 support.** You should reviiew HONE starting in the June 1995 time frame for any V3R1 image support performance information.

The implementation of ImagePlus Workfolder Application Facility/400 for accessing images stored in the QDLS file system folders is somewhat different between DOS-based and OS/2-based personal computers. You need to contact the image support personnel located in Gaithersburg, Maryland, USA, for information beyond the tips in this section and the product user documentation. Image support performance tips include:

- Keep the number of documents (images) within a single QDLS folder to 2000 or less.

  When the number of documents (can be images) within a single folder reaches approximately 2000, the speed of shared folder Reads and Writes may become significantly degraded.

- When using the Workfolder Application Facility/400 APIs, data area EKDAPILG must contain the character corresponding to the High Level Language (HLL) API being used for best performance.

  The following EKDAPILG data area characters are:

  – EKDAPILG = ′C′ (COBOL APIs)

  – EKDAPILG = ′R′ (RPG APIs)

  – EKDAPILG = ′O′ (Other APIs)

The default value is ′C.′ The principal performance impact is to keep the WAF database files open across returns from an API. Each API uses a different set of database files. If the data area does not contain the corresponding HLL character, it is possible several files will be closed and opened for each API call/return.

10.18 Ultimedia System Facilities/400

USF/400 is an optional feature of OS/400 that brings multimedia function to the AS/400. The AS/400 acts as a server ″data repository″ for image, video, and motion video data (object) that can be integrated into old and new applications that run in either the client workstation or the AS/400. The data is displayed on the client workstation.

USF/400 is a set of APIs that include repository and object services for capturing, registering, editing, sequencing and playing back the images within other application screen data.

The attached clients interface to USF/400 support by running either:

- OS/2 2.1 Multimedia Presentation Manager/2, or
- Microsoft Windows 3.1 with Multimedia Extensions (MME).

Client Access/400 is required and additional multimedia hardware and software may be required, depending on the type of multimedia data to be used and displayed.

The multimedia ″objects″ may be stored on the AS/400 in QDLS or the FSIOP. USF/400 requires QDLS for object tracking support even if the ″data″ is stored in the FSIOP.

Performance tips include:

- ″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 use of 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 FSIOP (LAN Server/400 required) is required in ″heavy video″ workloads, such as displaying a moving video.

  The FSIOP-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 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 9.13.3, “LAN Line and Control Unit Parameters” on page 173 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

Chapter 5 of the February 1995 *Performance Capabilities Reference* manual contains a large set of performance test results.

10.19 DataPropagator Relational/400

DataPropagator for the AS/400 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 would be appropriate include:

- Periodic replication of production data from a large AS/400 to a server model AS/400 for use by Decision Support users, perhaps in a GUI environment.
- Distribution of subset information to AS/400s 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 will guarantee safe delivery of the data.

As a general rule, DataPropagator Relational/400 is not a solution for those customers who expect instantaneous propagation of large amount of production data. Care should also be taken when selling it to customers with high CPU utilization (above 75%) on their AS/400 already.

Exercise caution in adding DataPropagator Relational/400 to an existing workload. The *Performance Capabilities Reference* manual contains performance information that should be reviewed periodically to determine availability of performance enhancements.

You can request this section on IBM VM by entering the following:

```
REQUEST DPROP400 FROM FIELDSIT AT RCHVMW2 (yourname
```

DataPropagator/400 Performance PTFs

Performance PTFs are being developed to reduce CPU impact of DataPropagator Relational/400. The first such PTF is SF22920, scheduled for availability in April 1995.

Refer to HONE item 130NC for general availability of performance PTFs.

10.19.1 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 defines the Distributed Relational Database (DRDB) parameters, authorized subscriptions (for example, the Copy Server systems) and the files/tables being changed. Journals must be defined as 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 which is used as a staging table for the apply process on the Copy Server systems.

The Copy Server system defines its DRDB parameters, and the frequency of change requests to send to the Data Server. It is the frequency of these requests and the complexity of the SQL statements that can impact CPU utilization on both the Data Server and the Copy Server systems.

The apply process runs on the Copy Server (″target system″) Using DRDA, the apply process either applies changes to the local copy (using the staging table from the Data Server as input) or it may refresh the local copy completely, using the original source table as input. The user can determine whether copies will be applied or the table will be 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.

10.19.1.1 Data Server Subsystems

```
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 41. DataPropagator/400 - Data Server Subsystem Jobs

For the Data Server system, the jobs QDPRCTL and QSQJRN perform the registration and data capture processes. 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.

10.19.1.2 Copy Server Subsystems

```
Subsystem QZSNDPR

   Job Name     User         Number   Type      Function

   QDPRAPPLY    SUBSCR       121307   BATCH     PGM-QZSAPLY
```

Figure 42. DataPropagator/400 - Copy Server Subsystem Jobs

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 via DDM job 129407 on the Data Server system.

Note that you may configure both the Data Server and the Copy Server functions on the same system. This technique could be used for testing and capacity planning purposes.

10.20 OptiConnect/400

OptiConnect/400 is an ″I″ listed PRPQ, that involves both software and hardware. It may be considered as a unique implementation of high speed DDM that is available for those customer environments where the single fastest AS/400 model available is not sufficient to contain the customer workload.

Each use of this PRPQ must be approved by the Rochester development lab as even with the high speed DDM implementation, the CPU utilization on the the client (without data) AS/400 and the serving (contains the database) AS/400 will be increased as a result of OptiConnect/400 work. The server system will also have increased disk I/O activity that could become significant if several client AS/400s are accessing the single server AS/400.

You must contact your IBM country AS/400 coordinator to begin the evaluation process to determine if the current customer configuration and database organization is a candidate for OptiConnect/400. The country coordinator will contact Rochester laboratory to determine the analysis process.

OptiConnect/400 database access performance is based on ″streamlined″ DDM driver code and use of the bus expansion adapters and bus optical cable as part of the PRPQ. This bus connection options have a maximum rated speed of 220M bits per second, though achievable rates will be significantly less for each customer application environment. RPQ numbers are used to support the various optic fiber cable lengths - 20, 60, 100, and up to 2000 meters for customized installations.

Note that since OptiConnect/400 uses DDM, remote data queues and data areas, and DB2/400 trigger support may be used. See index entries for these topics for additional information.

There are plans to make this PRPQ a licensed program sometime in 1995.

Appendix A. Guidelines for Interpreting Performance Data

Table 24. 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 (9332/9335/Internal Disk)	<0.40	0.40-0.50	>0.50
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. Need to query file QAPMIOPD to obtain these values. See index entry for *FSIOP* for sample queries.

Table 25. Machine Pool, Non-Database Page Faults

Main Storage Size	Good	Acceptable	Poor
All Systems	< 2	2-5	> 5

Table 26. Sum of Database and Non-Database Page Faults for Each Pool			
Model	**Good**	**Acceptable**	**Poor**
B10 B20 B30 B35 C04 C06 C10 D02 D04 E02	<10	10-15	>15
B40 B45 C20 C25 D06 D10 D20 D35 E04 E06 E10 F02 F04 2030	<15	15-25	>25
B50 B60 B70 D25 D45 D50 D60 E20 E25 E35 E45 E50 E60 F06 F10 F20 F25 F35 F45 F50 100 2010 2031 2032 2040 2041 2042	<25	25-50	>50
D70 D80 E70 E80 F60 F70 135 140 2411 2412 2043 2044 320/2050	<50	50-100	>100
E90 E95 F80	<100	100-200	>200
F90 F95 F97 2051 2052	<150	150-350	>350

Table 27. Sum of Database and Non-Database Page Faults in All Pools			
Model	**Good**	**Acceptable**	**Poor**
B10 B20 B30 B35 C04 C06 C10 D02 D04 E02	<15	15-25	>25
B40 B45 C20 C25 D06 D10 D20 D35 E04 E06 E10 F02 F04 2030	<25	25-40	>40
B50 B60 B70 D25 D45 D50 D60 E20 E25 E35 E45 E50 E60 F06 F10 F20 F25 F35 F45 F50 100 2010 2031 2032 2040 2041 2042	<35	35-60	>60
D70 D80 E70 E80 F60 F70 135 140 2411 2412 2043 2044 320/2050	<80	80-130	>130
E90 E95 F80	<180	180-300	>300
F90 F95 F97 2051 2052	<250	250-440	>440

Table 28. Ratio of Wait-to-Ineligible/Active-to-Wait		
Good	**Acceptable**	**Poor**
<.1	.1-.25	>.25

Table 29. Pool Size and Activity Level

Main Storage Size (MB)	Pool size (KB)/ Activity Level
4 - 12	500/2 - 1250/3
16 - 28	1500/4 - 2350/5
32 - 48	2700/5 - 4000/6
56 - 192	4625/5 - 14000/7
208 - 272	15000/8 - 18600/9
288 - 384	19500/9 - 23100/10
416 - 512	24000/15 - 27900/15
512 - 768	27900/15 - 37500/20
768 - 1024	37500/20 - 47100/20
1024 - 1280	47100/20 - 56000/25
1280 - 1536	56000/25 - 66300/30

Table 30. QINTER Activity Level Factor

Main Storage Size (MB)	Activity Level Factor (KB)
4 - 12	450
16 - 28	900
32 - 48	1600
64 - 192	2500
208 - 272	3000
288 - 384	3500
416 - 512	4000
512 - 768	4500
768 - 1024	5000
1024 - 1280	5500
1280 - 1536	6000

Table 31. QSPL Pool Sizes and Activity Levels for Advanced Function Printers

Number of Writers	Initial Size (KB)	Activity Levels
1	1500	1
2	1700	2
3	1900	3
4	2100	4
> 4	2300	5

Note: Note that if extensive use of AFP page segments and overlays are used set the storage pool to 4MB (4000KB).

Table 32. QSPL Pool Sizes and Activity Levels for Non-Advanced Function Printers		
Number of Writers	**Initial Size (KB)**	**Activity Level**
1	80	1
2	160	2
3	225	3
4	290	4
> 4	350	5

Table 33. BATCH Job Storage Guidelines		
Batch Job Type	**Initial Storage (KB)**	**Comments**
Short-Running Production	500	May run in 250KB; may require as much as 750KB
Long-Running Production	750	May run in 500KB; may require as much as 1000KB
Compiles	2000	May run in 1500KB; runs better in 3000KB
Reformat (Sort)	2000	Smaller sorts may run in 1500KB; larger sorts may use 2000 to 3000KB
Queries	2000	Smaller queries run in 1500KB; larger queries may use up to 4000KB
Save/Restore	2000	Some SAVE operations run in 1000KB; others may need 6000KB for maximum throughput.

Table 34. Authority Lookup - # of Exceptions vs. CPU % Cost - 1 of 6

Exceptions	B10	B20	B30	B35	B40	B45	B50	B60	B70	C04	C06	C10	C20	C25
10	5	2	3	3	2	2	1	0	0	5	3	3	2	2
25	12	7	9	8	7	5	3	2	1	12	9	9	7	6
50	25	14	19	16	14	11	7	4	3	25	19	19	14	12
100	50	29	39	33	28	22	15	9	7	50	39	39	28	24
200		59	78	67	56	44	30	19	14		78	78	56	48
300		89			84	66	46	29	22				84	72
400						88	61	39	29					97
500							77	49	37					
600							92	59	44					
700								69	52					
800								79	59					
900								89	67					
1000								99	74					

Note: The CPU usage values are based on a worst case scenario of instruction mixes and CPU times and may not match your particular situation or installation.

Table 35. Authority Lookup - # of Exceptions vs. CPU % Cost - 2 of 6												
Exceptions	**D02**	**D04**	**D06**	**D10**	**D20**	**D25**	**D35**	**D45**	**D50**	**D60**	**D70**	**D80**
10	3	3	2	2	2	1	2	1	1	0	0	0
25	9	8	6	7	5	3	5	3	2	1	1	0
50	19	16	13	14	11	7	10	6	5	3	2	1
100	39	33	26	28	22	15	20	13	11	6	4	2
200	78	67	53	56	44	30	40	27	22	12	9	5
300			80	84	66	46	61	41	33	18	14	8
400					88	61	81	55	45	25	18	10
500						77		68	56	31	23	13
600						92		82	67	37	28	16
700								96	79	44	32	18
800									90	50	37	21
900										56	42	24
1000										62	44	26
2000											93	53
3000												80

Table 36. Authority Lookup - # of Exceptions vs. CPU % Cost - 3 of 6														
Exceptions	**E02**	**E04**	**E06**	**E10**	**E20**	**E25**	**E35**	**E45**	**E50**	**E60**	**E70**	**E80**	**E90**	**E95**
10	3	2	2	1	1	1	1	1	0	0	0	0	0	0
25	8	6	5	4	3	3	3	2	2	1	0	0	0	0
50	16	13	10	9	7	6	7	5	4	2	1	1	0	0
100	33	26	20	19	15	12	15	10	8	5	3	2	1	1
200	67	53	40	39	30	25	30	21	16	10	7	4	3	2
300		80	61	58	46	38	46	32	24	15	11	6	4	3
400			81	78	61	50	61	43	32	21	15	8	6	5
500				98	77	63	77	54	41	26	18	10	7	6
600					92	76	92	65	49	31	22	12	9	7
700						89		75	57	36	26	14	10	8
800								86	65	42	30	17	12	10
900								97	74	47	33	19	14	11
1000									82	52	37	21	15	12
2000											75	42	31	25
3000												64	46	38
4000												85	62	50
5000													78	63
6000													94	76
7000														88

Table 37. Authority Lookup - # of Exceptions vs. CPU % Cost - 4 of 6

Exceptions	F02	F04	F06	F10	F20	F25	F35	F45	F50	F60	F70	F80	F90	F95	F97
10	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0
25	6	5	3	3	3	2	2	2	1	0	0	0	0	0	0
50	13	10	7	7	6	5	5	4	2	1	1	0	0	0	0
100	26	20	15	15	12	10	10	8	5	3	2	1	1	1	0
200	53	40	30	30	25	21	21	17	10	7	5	3	2	1	1
300	80	61	46	46	38	32	32	25	16	11	8	4	3	2	2
400		81	61	61	50	43	43	34	21	14	10	6	4	3	3
500			77	77	63	54	54	43	26	18	13	7	5	4	4
600			92	92	76	65	65	51	32	22	16	9	6	5	4
700					89	75	75	60	37	26	18	10	8	6	5
800						86	86	69	42	29	21	12	9	7	6
900						97	97	77	48	33	24	13	10	8	7
1000								86	53	37	26	15	11	9	8
2000										74	53	30	23	19	16
3000											80	45	34	29	24
4000												60	46	39	33
5000												76	58	49	41
6000												91	70	59	50
7000													81	69	58
8000													92	79	66
9000														89	74
12000															99

Table 38. Authority Lookup - # of Exceptions vs. CPU % Cost - 5 of 6

Exceptions	2030	2031	2032	2040	2041	2042	2043	2044	2050	2051	2052
10	2	1	0	1	0	0	0	0	0	0	0
25	5	3	2	3	2	1	1	0	0	0	0
50	10	6	4	6	4	3	2	1	1	0	0
100	20	12	8	12	8	7	4	2	2	1	0
200	40	25	17	25	17	14	9	4	2	1	
300	61	38	26	38	26	21	14	8	6	3	2
400	81	50	35	50	35	28	18	10	8	4	3
500		63	43	63	43	35	23	13	10	6	4
600		76	52	76	52	42	28	16	13	7	4
700		89	61	89	61	49	32	18	15	8	5
800			70		70	56	37	21	17	9	6
900			79		79	63	42	24	19	11	7
1000			87		87	70	46	26	21	12	8
2000							93	53	43	24	16
3000								80	65	37	24
4000									87	49	33
5000										61	41
7000										86	58
9000											74
12000											99

Table 39. Authority Lookup - # of Exceptions vs. CPU % Cost - 6 of 6												
Exceptions	**100i**	**135i**	**140i**	**100n**	**135n**	**140n**	**2010i**	**2411i**	**2412i**	**2010n**	**2411n**	**2412n**
10	2	1	1	0	0	0	2	1	1	0	0	0
25	6	3	3	2	1	0	6	3	3	2	1	0
50	13	7	6	4	2	1	13	7	6	4	2	1
100	26	15	12	8	4	2	26	15	12	8	4	2
200	53	30	25	17	9	4	53	30	25	17	9	4
300	80	46	38	25	14	6	80	46	38	25	14	6
400		61	50	34	18	9		61	50	34	18	8
500		77	63	43	23	11		77	63	43	23	10
600		92	76	51	28	13		92	76	51	28	13
700			89	60	32	15			89	60	32	15
800				69	37	18				69	37	17
900				77	42	20				77	42	19
1000				86	46	22				86	46	21
2000					93	45					93	43
3000						67						65
4000						90						86
Note: i = Interactive n = Non-Interactive												

Table 40. EAO - # of Exceptions vs. CPU % Cost - 1 of 6														
Exceptions	**B10**	**B20**	**B30**	**B35**	**B40**	**B45**	**B50**	**B60**	**B70**	**C04**	**C06**	**C10**	**C20**	**C25**
100	5	3	4	3	3	2	1	1	0	5	4	4	3	2
200	11	6	9	7	6	5	3	2	1	11	9	9	6	5
300	17	10	13	11	9	7	5	3	2	17	13	13	9	8
400	23	13	18	15	13	10	7	4	3	23	18	18	13	11
500	29	17	22	19	16	12	8	5	4	29	22	22	16	14
600	35	20	27	23	19	15	10	6	5	35	27	27	19	16
700	41	24	31	27	22	17	12	8	6	41	31	31	22	19
800	47	27	36	31	26	20	14	9	6	47	36	36	26	22
900	52	31	40	35	29	23	16	10	7	52	40	40	29	25
1000	58	34	45	39	32	25	17	11	8	58	45	45	32	28
2000		69	90	78	65	51	35	23	17		90	90	65	56
3000					98	76	53	34	25				98	84
4000							71	46	34					
5000							89	57	43					
10000									86					

Note: The CPU usage values are based on a worst case scenario of instruction mixes and CPU times and may not match your particular situation or installation.

Table 41. EAO - # of Exceptions vs. CPU % Cost - 2 of 6												
Exceptions	**D02**	**D04**	**D06**	**D10**	**D20**	**D25**	**D35**	**D45**	**D50**	**D60**	**D70**	**D80**
100	4	3	3	3	2	1	2	1	1	0	0	0
200	9	7	6	6	5	3	4	3	2	1	1	0
300	13	11	9	9	7	5	7	4	3	2	1	0
400	18	15	12	13	10	7	9	6	5	2	2	1
500	22	19	15	16	12	8	11	7	6	3	2	1
600	27	23	18	19	15	10	14	9	7	4	3	1
700	31	27	21	22	17	12	16	11	9	5	3	2
800	36	31	24	26	20	14	18	12	10	5	4	2
900	40	35	27	29	23	16	21	14	11	6	4	2
1000	45	39	30	32	25	17	23	15	13	7	5	3
2000	90	78	61	65	51	35	47	31	26	14	10	6
3000			92	98	76	53	70	47	39	21	16	9
4000						71	94	63	52	29	21	12
5000						89		79	65	36	26	15
10000										72	53	30
20000												61
30000												92

Exceptions	E02	E04	E06	E10	E20	E25	E35	E45	E50	E60	E70	E80	E90	E95
100	3	3	2	2	1	1	1	1	0	0	0	0	0	0
200	7	6	4	4	3	2	3	2	1	1	0	0	0	0
300	11	9	7	6	5	4	5	3	2	1	1	0	0	0
400	15	12	9	9	7	5	7	5	3	2	1	0	0	0
500	19	15	11	11	8	7	8	6	4	3	2	1	0	0
600	23	18	14	13	10	8	10	7	5	3	2	1	1	0
700	27	21	16	15	12	10	12	8	6	4	3	1	1	1
800	31	24	18	18	14	11	14	10	7	4	3	1	1	1
900	35	27	21	20	16	13	16	11	8	5	3	2	1	1
1000	39	30	23	22	17	14	17	12	9	6	4	2	1	1
2000	78	61	47	45	35	29	35	25	18	12	8	4	3	2
3000		92	70	67	53	44	53	37	28	18	13	7	5	4
4000			94	90	71	58	71	50	37	24	17	9	7	5
5000					89	73	89	62	47	30	21	12	9	7
10000									94	60	43	24	18	14
20000											87	49	36	29
30000												74	54	43
40000												98	72	58
45000													81	65
50000													90	73
60000														88

Table 42. EAO - # of Exceptions vs. CPU % Cost - 3 of 6

Table 43. EAO - # of Exceptions vs. CPU % Cost - 4 of 6

Exceptions	F02	F04	F06	F10	F20	F25	F35	F45	F50	F60	F70	F80	F90	F95	F97
100	3	2	1	1	1	1	1	0	0	0	0	0	0	0	0
200	6	4	3	3	2	2	2	1	1	0	0	0	0	0	0
300	9	7	5	5	4	3	3	2	1	1	0	0	0	0	0
400	12	9	7	7	5	5	5	3	2	1	1	0	0	0	0
500	15	11	8	8	7	6	6	4	3	2	1	0	0	0	0
600	18	14	10	10	8	7	7	5	3	2	1	1	0	0	0
700	21	16	12	12	10	8	8	6	4	3	2	1	0	0	0
800	24	18	14	14	11	10	10	7	4	3	2	1	1	0	0
900	27	21	16	16	13	11	11	8	5	3	2	1	1	1	0
1000	30	23	17	17	14	12	12	9	6	4	3	1	1	1	0
2000	61	47	35	35	29	25	25	19	12	8	6	3	2	2	1
3000	92	70	53	53	44	37	37	29	18	12	9	5	4	3	2
4000		94	71	71	58	50	50	39	24	17	12	7	5	4	3
5000			89	89	73	62	62	49	30	21	15	8	6	5	4
10000								99	61	42	31	17	13	11	9
20000										85	62	35	26	23	19
30000											93	52	40	34	28
40000												70	53	46	38
45000												79	60	51	43
50000												87	67	57	48
60000													80	70	58
70000													94	80	67
80000														92	77
90000															86

Table 44. EAO - # of Exceptions vs. CPU % Cost - 5 of 6											
Exceptions	**2030**	**2031**	**2032**	**2040**	**2041**	**2042**	**2043**	**2044**	**2050**	**2051**	**2052**
100	2	1	1	1	1	0	0	0	0	0	0
200	4	2	2	2	2	1	1	0	0	0	0
300	7	4	3	4	3	2	1	0	0	0	0
400	9	5	4	5	4	3	2	1	1	0	0
500	11	7	5	7	5	4	2	1	1	0	0
600	14	8	6	8	6	4	3	1	1	0	0
700	16	10	7	10	7	5	3	2	1	0	0
800	18	11	8	11	8	6	4	2	2	1	0
900	21	13	9	13	9	7	4	2	2	1	0
1000	23	14	10	14	10	8	5	3	2	1	0
2000	47	29	20	29	20	16	10	6	5	2	1
3000	70	44	30	44	30	24	16	9	7	4	2
4000	94	58	40	58	40	32	21	12	10	5	3
5000		73	50	73	50	40	26	15	12	7	4
10000						81	53	30	25	14	9
20000								61	50	28	19
30000								92	76	42	38
40000										56	38
45000										64	43
50000										71	48
70000										99	67
90000											86

Table 45. EAO - # of Exceptions vs. CPU % Cost - 6 of 6

Exceptions	100i	135i	140i	100n	135n	140n	2010i	2411i	2412i	2010n	2411n	2412n
100	3	1	1	0	0	0	3	1	1	0	0	0
200	6	3	2	1	1	0	6	3	2	1	1	0
300	9	5	4	2	1	0	9	5	4	2	1	0
400	12	7	5	3	2	1	12	7	5	3	2	1
500	15	8	7	4	2	1	15	8	7	4	2	1
600	18	10	8	5	3	1	18	10	8	5	3	1
700	21	12	10	6	3	1	21	12	10	6	3	1
800	24	14	11	7	4	2	24	14	11	7	4	2
900	27	16	13	8	4	2	27	16	13	8	4	2
1000	30	17	14	9	5	2	30	17	14	9	5	2
2000	61	35	29	19	10	5	61	35	29	19	10	5
3000	92	53	44	29	16	7	92	53	44	29	16	7
4000		71	58	39	21	10		71	58	39	21	10
5000		89	73	49	26	13		89	73	49	26	12
10000				99	53	26				99	53	25
20000						52						50
30000						78						75

Note: i = Interactive n = Non-Interactive

Table 46. Size - # of Exceptions vs. CPU % Cost - 1 of 6

Exceptions	B10	B20	B30	B35	B40	B45	B50	B60	B70	C04	C06	C10	C20	C25
25	4	2	3	2	2	1	1	0	0	4	3	3	2	1
50	8	4	6	5	4	3	2	1	1	8	6	6	4	3
100	16	9	12	11	9	7	5	3	2	16	12	12	9	7
200	33	19	25	22	18	14	10	6	4	33	25	25	18	15
400	66	39	51	44	36	28	20	13	9	66	51	51	36	31
600	99	58	76	66	55	43	30	19	14	99	76	76	55	47
800		78		88	73	57	40	26	19				73	63
1000		97			92	72	50	32	24				92	79
2000								65	48					
3000								97	73					
4000									97					

Note: The CPU usage values are based on a worst case scenario of instruction mixes and CPU times and may not match your particular situation or installation.

Table 47. Size - # of Exceptions vs. CPU % Cost - 2 of 6

Exceptions	D02	D04	D06	D10	D20	D25	D35	D45	D50	D60	D70	D80
25	3	2	2	2	1	1	1	1	0	0	0	0
50	6	5	4	4	3	2	3	2	1	1	0	0
100	12	11	8	9	7	5	6	4	3	2	1	0
200	25	22	17	18	14	10	13	8	7	4	3	1
400	51	44	34	36	28	20	26	17	14	8	6	3
600	76	66	52	55	43	30	39	26	22	12	9	5
800		88	69	73	57	40	53	35	29	16	12	6
1000			87	92	72	50	66	44	36	20	15	8
2000								89	73	40	30	17
3000										61	45	26
4000										81	60	34
5000											76	43
10000												87

Table 48. Size - # of Exceptions vs. CPU % Cost - 3 of 6														
Exceptions	**E02**	**E04**	**E06**	**E10**	**E20**	**E25**	**E35**	**E45**	**E50**	**E60**	**E70**	**E80**	**E90**	**E95**
25	2	2	1	1	1	1	1	0	0	0	0	0	0	0
50	5	4	3	3	2	2	2	1	1	0	0	0	0	0
100	11	8	6	6	5	4	5	3	2	1	1	0	0	0
200	22	17	13	12	10	8	10	7	5	3	2	1	1	0
400	44	34	26	25	20	16	20	14	10	6	4	2	2	1
600	66	52	39	38	30	24	30	21	16	10	7	4	3	2
800	88	69	53	51	40	33	40	28	21	13	9	5	4	3
1000		87	66	63	50	41	50	35	26	17	12	6	5	4
2000						82		70	53	34	24	13	10	8
3000									80	51	36	20	15	12
4000										68	49	27	20	16
5000										85	61	34	25	20
10000												69	50	41
15000													76	61
20000														82

Table 49. Size - # of Exceptions vs. CPU % Cost - 4 of 6															
Exceptions	**F02**	**F04**	**F06**	**F10**	**F20**	**F25**	**F35**	**F45**	**F50**	**F60**	**F70**	**F80**	**F90**	**F95**	**F97**
25	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0
50	4	3	2	2	2	1	1	1	0	0	0	0	0	0	0
100	8	6	5	5	4	3	3	2	1	1	0	0	0	0	0
200	17	13	10	10	8	7	7	5	3	2	1	0	0	0	0
400	34	26	20	20	16	14	14	11	6	4	3	1	1	1	1
600	52	39	30	30	24	21	21	16	10	7	5	2	2	1	1
800	69	53	40	40	33	28	28	22	13	9	7	3	3	2	2
1000	87	66	50	50	41	35	35	28	17	12	8	4	3	3	2
2000					82	70	70	56	34	24	17	9	7	6	5
3000								84	52	36	26	14	11	9	8
4000									69	48	35	19	15	13	10
5000									87	60	43	24	18	16	13
10000											87	49	37	32	27
15000												74	56	48	40
20000												99	75	65	54
25000													94	81	67
30000														97	81
35000															94

Exceptions	2030	2031	2032	2040	2041	2042	2043	2044	2050	2051	2052
25	1	1	0	1	0	0	0	0	0	0	0
50	3	2	1	2	1	1	0	0	0	0	0
100	6	4	2	4	2	2	1	0	0	0	0
200	13	8	5	8	5	4	3	1	1	0	0
400	26	16	11	16	11	9	6	3	2	1	1
600	39	24	17	24	17	13	9	5	4	2	1
800	53	33	22	33	22	18	12	6	5	3	2
1000	66	41	28	41	28	23	15	8	7	4	2
2000		82	57	82	57	46	30	17	14	8	5
3000			85		85	69	45	26	21	12	8
4000						92	60	34	28	16	10
5000							76	43	35	20	13
10000								87	71	40	27
15000										60	40
20000										80	54
25000											67
30000											81
35000											94

Table 50. Size - # of Exceptions vs. CPU % Cost - 5 of 6

Exceptions	100i	135i	140i	100n	135n	140n	2010i	2411i	2412i	2010n	2411n	2412n
25	2	1	1	0	0	0	2	1	1	0	0	0
50	4	2	2	1	0	0	4	2	2	1	0	0
100	8	5	4	2	1	0	8	5	4	2	1	0
200	17	10	8	5	3	1	17	10	8	5	3	1
400	34	20	16	11	6	2	34	20	16	11	6	2
600	52	30	24	16	9	4	52	30	24	16	9	4
800	69	40	33	22	12	5	69	40	33	22	12	5
1000	87	50	41	28	15	7	87	50	41	28	15	7
2000			82	56	30	14			82	56	30	14
3000				84	45	22				84	45	21
4000					60	29					60	28
5000					76	36					76	35
10000						73						70

Table 51. Size - # of Exceptions vs. CPU % Cost - 6 of 6

Note: i = Interactive n = Non-Interactive

Table 52. Verify - # of Exceptions vs. CPU % Cost - 1 of 6

Exceptions	B10	B20	B30	B35	B40	B45	B50	B60	B70	C04	C06	C10	C20	C25
10	6	4	5	4	3	2	2	1	1	6	5	5	3	3
25	17	10	13	11	9	7	5	3	2	17	13	13	9	8
50	34	20	26	22	19	14	10	6	5	34	26	26	19	16
100	68	40	52	45	38	29	20	13	10	68	52	52	38	32
200		80		91	76	59	41	26	20				76	65
300						89	62	40	30					98
400							83	53	40					
500								67	50					
600								80	60					
700								94	70					
800									80					
900									91					

Note: The CPU usage values are based on a worst case scenario of instruction mixes and CPU times and may not match your particular situation or installation.

Table 53. Verify - # of Exceptions vs. CPU % Cost - 2 of 6

Exceptions	D02	D04	D06	D10	D20	D25	D35	D45	D50	D60	D70	D80
10	5	4	3	3	2	2	2	1	1	0	0	0
25	13	11	9	9	7	5	6	4	3	2	1	0
50	26	22	18	19	14	10	13	9	7	4	3	1
100	52	45	36	38	29	20	27	18	15	8	6	3
200		91	72	76	59	41	55	37	30	16	12	7
300					89	62	82	55	45	25	18	10
400						83		74	61	33	25	14
500								92	76	42	31	18
600									91	50	37	21
700										59	44	25
800										67	50	28
900										76	56	32
1000										84	63	36
2000												72

Table 54. Verify - # of Exceptions vs. CPU % Cost - 3 of 6

Exceptions	E02	E04	E06	E10	E20	E25	E35	E45	E50	E60	E70	E80	E90	E95
10	4	3	2	2	2	1	2	1	1	0	0	0	0	0
25	11	9	6	6	5	4	5	3	2	1	1	0	0	0
50	22	18	13	13	10	8	10	7	5	3	2	1	1	0
100	45	36	27	26	20	17	20	14	11	7	5	2	2	1
200	91	72	55	52	41	34	41	29	22	14	10	5	4	3
300			82	79	62	51	62	43	33	21	15	8	6	5
400					83	68	83	58	44	28	20	11	8	6
500						85		73	55	35	25	14	10	8
600								87	66	42	30	17	12	10
700									77	49	35	20	14	11
800									88	56	40	23	16	13
900									99	63	45	26	18	15
1000										70	50	28	21	17
2000												57	42	34
4000													84	68
5000														85

Table 55. Verify - # of Exceptions vs. CPU % Cost - 4 of 6

Exceptions	F02	F04	F06	F10	F20	F25	F35	F45	F50	F60	F70	F80	F90	F95	F97
10	3	2	2	2	1	1	1	1	0	0	0	0	0	0	0
25	9	6	5	5	4	3	3	2	1	1	0	0	0	0	0
50	18	13	10	10	8	7	7	5	3	2	1	1	0	0	0
100	36	27	20	20	17	14	14	11	7	5	3	2	1	1	1
200	72	55	41	41	34	29	29	23	14	10	7	4	3	2	2
300		82	62	62	51	43	43	34	21	15	10	6	4	4	3
400			83	83	68	58	58	46	28	20	14	8	6	5	4
500					85	73	73	58	36	25	18	10	7	6	5
600						87	87	69	43	30	21	12	9	8	6
700								81	50	35	25	14	10	9	7
800								93	57	40	29	16	12	10	8
900									65	45	32	18	14	12	10
1000									72	50	36	20	15	13	11
2000											72	41	31	26	22
3000												61	47	40	33
4000												82	62	53	44
5000													78	67	56
6000													94	80	67
7000														94	78
8000															89

Table 56. Verify - # of Exceptions vs. CPU % Cost - 5 of 6

Exceptions	**2030**	**2031**	**2032**	**2040**	**2041**	**2042**	**2043**	**2044**	**2050**	**2051**	**2052**
10	2	1	1	1	1	0	0	0	0	0	0
25	6	4	2	4	2	2	1	0	0	0	0
50	13	8	5	8	5	4	3	1	1	0	0
100	27	17	11	17	11	9	6	3	2	1	1
200	55	34	23	34	23	19	12	7	5	3	2
300	82	51	35	51	35	28	18	10	8	4	3
400		68	47	68	47	38	25	14	11	6	4
500		85	59	85	59	47	31	18	14	8	5
600			71		71	57	37	21	17	9	6
700			83		83	66	44	25	20	11	7
800			94		94	76	50	28	23	13	8
900						85	56	32	26	14	10
1000						95	63	36	29	16	11
2000								72	59	33	22
3000									88	49	33
4000										66	44
5000										83	56
6000										99	67
7000											78
8000											89

Table 57. Verify - # of Exceptions vs. CPU % Cost - 6 of 6

Exceptions	**100i**	**135i**	**140i**	**100n**	**135n**	**140n**	**2010i**	**2411i**	**2412i**	**2010n**	**2411n**	**2412n**
10	3	2	1	1	0	0	3	2	1	1	0	0
25	9	5	4	2	1	0	9	5	4	2	1	0
50	18	10	8	5	3	1	18	10	8	5	3	1
100	36	20	17	11	6	3	36	20	17	11	6	2
200	72	41	34	23	12	6	72	41	34	23	12	5
300		62	51	34	18	9		62	51	34	18	8
400		83	68	46	25	12		83	68	46	25	11
500			85	58	31	15			85	58	31	14
600				69	37	18				69	37	17
700				81	44	21				81	44	20
800				93	50	24				93	50	23
900					56	27					56	26
1000					63	30					63	29
2000						61						58
3000						91						87

Note: i = Interactive n = Non-Interactive

Table 58. Decimal Data - # of Exceptions vs. CPU % Cost - 1 of 6

Exceptions	B10	B20	B30	B35	B40	B45	B50	B60	B70	C04	C06	C10	C20	C25
25	4	2	3	3	2	2	1	0	0	4	3	3	2	2
50	9	5	7	6	5	4	2	1	1	9	7	7	5	4
100	18	10	14	12	10	8	5	3	2	18	14	14	10	8
200	37	21	28	24	20	16	11	7	5	37	28	28	20	17
400	74	43	57	49	41	32	22	14	10	74	57	57	41	35
600		65	85	74	61	48	33	21	16		85	85	61	52
800		87		98	82	64	44	29	21				82	70
1000						80	56	36	27					88
2000								72	54					
3000									81					

Note: The CPU usage values are based on a worst case scenario of instruction mixes and CPU times and may not match your particular situation or installation.

Table 59. Decimal Data - # of Exceptions vs. CPU % Cost - 2 of 6

Exceptions	D02	D04	D06	D10	D20	D25	D35	D45	D50	D60	D70	D80
25	3	3	2	2	2	1	1	1	1	0	0	0
50	7	6	4	5	4	2	3	2	2	1	0	0
100	14	12	9	10	8	5	7	5	4	2	1	0
200	28	24	19	20	16	11	14	10	8	4	3	1
400	57	49	39	41	32	22	29	20	16	9	6	3
600	85	74	58	61	48	33	44	30	24	13	10	5
800		98	78	82	64	44	59	40	32	18	13	7
1000			97		80	56	74	50	41	22	17	9
2000									82	45	34	19
3000										68	51	29
4000										91	68	39
5000											85	48
8000												78
10000												97

Table 60. Decimal Data - # of Exceptions vs. CPU % Cost - 3 of 6														
Exceptions	**E02**	**E04**	**E06**	**E10**	**E20**	**E25**	**E35**	**E45**	**E50**	**E60**	**E70**	**E80**	**E90**	**E95**
25	3	2	1	1	1	1	1	0	0	0	0	0	0	0
50	6	4	3	3	2	2	2	1	1	0	0	0	0	0
100	12	9	7	7	5	4	5	3	2	1	1	0	0	0
200	24	19	14	14	11	9	11	7	5	3	2	1	1	0
400	49	39	29	28	22	18	22	15	11	7	5	3	2	1
600	74	58	44	42	33	27	33	23	17	11	8	4	3	2
800	98	78	59	57	44	37	44	31	23	15	10	6	4	3
1000		97	74	71	56	46	56	39	29	19	13	7	5	4
2000						92		78	59	38	27	15	11	9
3000									89	57	41	23	17	13
4000										76	54	31	22	18
5000										95	68	38	28	23
8000												62	45	36
10000												77	56	46
15000													85	69
20000														92

Table 61. Decimal Data - # of Exceptions vs. CPU % Cost - 4 of 6															
Exceptions	**F02**	**F04**	**F06**	**F10**	**F20**	**F25**	**F35**	**F45**	**F50**	**F60**	**F70**	**F80**	**F90**	**F95**	**F97**
25	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0
50	4	3	2	2	2	1	1	1	0	0	0	0	0	0	0
100	9	7	5	5	4	3	3	3	1	1	0	0	0	0	0
200	19	14	11	11	9	7	7	6	3	2	1	1	0	0	0
400	39	29	22	22	18	15	15	12	7	5	3	2	1	1	1
600	58	44	33	33	27	23	23	18	11	8	5	3	2	2	1
800	78	59	44	44	37	31	31	25	15	10	7	4	3	2	2
1000	97	74	56	56	46	39	39	31	19	13	9	5	4	3	3
2000					92	78	78	62	39	27	19	11	8	7	6
3000								94	58	40	29	16	12	10	9
4000									78	54	39	22	16	14	12
5000									97	67	49	27	21	18	15
8000											78	44	33	29	24
10000											98	55	42	36	30
15000												82	63	54	45
20000													84	72	60
25000														90	75
30000															90

Table 62. Decimal Data - # of Exceptions vs. CPU % Cost - 5 of 6											
Exceptions	**2030**	**2031**	**2032**	**2040**	**2041**	**2042**	**2043**	**2044**	**2050**	**2051**	**2052**
25	1	1	0	1	0	0	0	0	0	0	0
50	3	2	1	2	1	1	0	0	0	0	0
100	7	4	3	4	3	2	1	0	0	0	0
200	14	9	6	9	6	5	3	1	1	0	0
400	29	18	12	18	12	10	6	3	3	1	1
600	44	27	19	27	19	15	10	5	4	2	1
800	59	37	25	37	25	20	13	7	6	3	2
1000	74	46	31	46	31	25	17	9	7	4	3
2000		92	63	92	63	51	34	19	15	8	6
3000			95		95	77	51	29	23	13	9
4000							68	39	31	17	12
5000							85	48	39	22	15
8000								78	63	35	24
10000								97	79	44	30
15000										67	45
20000										89	60
25000											75
30000											90

Table 63. Decimal Data - # of Exceptions vs. CPU % Cost - 6 of 6												
Exceptions	**100i**	**135i**	**140i**	**100n**	**135n**	**140n**	**2010i**	**2411i**	**2412i**	**2010n**	**2411n**	**2412n**
25	2	1	1	0	0	0	2	1	1	0	0	0
50	4	2	2	1	0	0	4	2	2	1	0	0
100	9	5	4	3	1	0	9	5	4	3	1	0
200	19	11	9	6	3	1	19	11	9	6	3	1
400	39	22	18	12	6	3	39	22	18	12	6	3
600	58	33	27	18	10	4	58	33	27	18	10	4
800	78	44	37	25	13	6	78	44	37	25	13	6
1000	97	56	46	31	17	8	97	56	46	31	17	7
2000			92	62	34	16			92	62	34	15
3000				94	51	24				94	51	23
4000					68	32					68	31
5000					85	41					85	39
8000						65						63
10000						82						78
Note: i = Interactive n = Non-Interactive												

Table 64. AS/400 Model Comparison Summary - 1 of 4									
AS/400 9402 Model	**Internal Relative Processor Performance**	**Power over E Model**	**Memory Range (MB)**	**Disk Capacity Max (GB)**	**Comm Lines**	**LAN Lines**	**Max Twinax WS**	**Max ASCII WS**	**CPU Sec per Tran Guide**
C04	1.0		8-12	.6 - 1.3	1-5	0-1	14	6	<.60
C06	1.3		8-16	.6 - 1.3	1-5	0-1	54	24	<.60
D02	1.3		8-16	.6 - 1.2	0-3	0-1	14	12	<.60
D04	1.5		8-16	.8 - 1.6	1-8	0-1	28	12	<.45
D06	1.9		8-20	.8 - 1.6	1-8	0-1	54	24	<.45
E02	1.5		8-24	1.0 - 2.0	0-3	0-1	14	12	<.45
E04	1.9		8-24	1.0 - 4.0	1-8	0-1	42	48	<.45
E06	2.5		8-40	1.0 - 7.9	1-8	0-1	68	66	<.40
F02	1.9	25%	8-24	1.0 - 2.0	0-8	0-1	28	18	<.45
F04	2.5	30%	8-24	1.0 - 4.0	1-8	0-1	68	66	<.40
F06	3.3	25%	8-40	1.0 - 7.9	1-14	0-2	108	102	<.30

Table 65. AS/400 Model Comparison Summary - 2 of 4									
AS/400 9404 Model	**Internal Relative Processor Performance**	**Power over E Model**	**Memory Range (MB)**	**Disk Capacity Range (GB)**	**Comm Lines**	**LAN Lines**	**Max Twinax WS**	**Max ASCII WS**	**CPU Sec per Tran Guide**
B10	1.0		4-16	.6 - 1.9	1-6	0-1	40	36	<.60
B20	1.7		4-28	.6 - 3.8	1-14	0-1	80	72	<.45
C10	1.3		8-20	.6 - 1.9	1-8	0-1	40	36	<.60
C20	1.8		8-32	.6 - 3.8	1-8	0-1	80	72	<.45
C25	2.1		8-40	.6 - 3.8	1-14	0-2	80	72	<.40
D10	1.8		8-32	.8 - 4.8	1-14	0-2	80	72	<.45
D20	2.3		8-40	.8 - 4.8	1-14	0-2	80	72	<.40
D25	3.3		16-64	.8 - 6.4	1-14	0-2	160	108	<.30
E10	2.6		8-40	1.0 - 11.9	1-14	0-2	160	162	<.40
E20	3.3		8-72	1.0 - 11.9	1-20	0-2	160	162	<.30
E25	4.0		16-80	1.0 - 15.8	1-26	0-3	240	162	<.30
F10	3.3	30%	8-72	1.0 - 11.8	1-14	0-2	160	162	<.30
F20	4.0	20%	16-80	1.0 - 11.8	1-20	0-4	240	162	<.25
F25	4.7	15%	16-80	1.0 - 15.8	1-26	0-4	240	162	<.25

Table 66. AS/400 Model Comparison Summary - 3 of 4									
AS/400 9406 Model	**Internal Relative Processor Performance**	**Power over E Model**	**Memory Range (MB)**	**Disk Capacity Range (GB)**	**Comm Lines**	**LAN Lines**	**Max Twinax WS**	**Max ASCII WS**	**CPU Sec per Tran Guide**
B30	1.3		4-36	.6 - 13.7	1-14	0-2	160	72	<.45
B35	1.5		8-40	.6 - 13.7	1-16	0-4	160	72	<.45
B40	1.8		8-40	.6 - 13.7	1-14	0-2	240	108	<.30
B45	2.3		8-40	.6 - 13.7	1-32	0-4	240	180	<.30
B50	3.3		16-48	.6 - 27.4	1-32	0-4	400	108	<.30
B60	5.1		32-96	.6 - 54.8	1-32	0-4	600	270	<.25
B70	6.8		32-192	.6 - 54.8	1-48	0-4	800	360	<.20
D35	2.5		8-72	1.3 - 67	1-16	0-4	240	108	<.40
D45	3.7		16-80	1.3 - 67	1-32	0-4	400	180	<.30
D50	4.5		32-128	1.3 - 98.0	1-32	0-4	600	270	<.25
D60	8.1		64-192	1.3 - 146	1-32	0-4	800	360	<.25
D70	10.9		64-256	1.3 - 146	1-48	0-4	1200	540	<.20
D80	19.0 (2)		64-384	1.3 - 256	1-64	0-4	2000	900	<.20
E35	3.3		8-72	1.3 - 67	1-20	0-4	360	162	<.30
E45	4.7		16-80	1.3 - 67	1-33	0-4	480	216	<.25
E50	6.2		32-128	1.3 - 98.0	1-33	0-4	720	324	<.20
E60	9.7		64-192	1.3 - 146	1-33	0-4	1000	450	<.20
E70	13.5		64-256	1.3 - 146	1-49	0-4	1400	630	<.20
E80	23.8 (2)		64-512	1.3 - 256	1-64	0-6	2400	1080	<.15
E90	32.6 (3)		64-1024	1.3 - 256	1-64	0-6	2400	1080	<.15
E95	40.2 (4)		64-1152	1.3 - 256	1-64	0-6	2400	1080	<.15
F35	4.7	40%	16-80	2.0 - 67	1-20	0-4	480	216	<.25
F45	5.9	25%	16-80	2.0 - 67	1-33	0-4	720	324	<.20
F50	9.5	60%	64-192	2.0 - 114	1-33	0-4	1000	450	<.20
F60	13.7	40%	128-384	2.0 - 146	1-49	0-4	1400	630	<.20
F70	18.9	45%	128-512	2.0 - 256	1-64	0-6	2400	1080	<.15
F80	33.5 (2)	40%	128-768	2.0 - 252	1-64	0-6	2400	1080	<.15
F90	43.8 (3)	35%	128-1024	2.0 - 256	1-64	0-6	2400	1080	<.15
F95	51.0 (4)	30%	128-1280	2.0 - 256	1-64	0-6	2400	1080	<.15
F97	61.2 (4)	30%	128-1536	2.0 - 256	1-64	0-6	4800	2160	<.15

Note: Relative Internal Processor Performance based on **V2R2** standardized instruction mix. (n) indicates the number of processors. No disk constraints. CPU seconds per transaction guideline based on anticipated increase in transaction complexity (processing) on larger model CPUs.

Refer to *V3R1 Performance Capabilities Reference*, ZC41-8166-3, for Relative Internal Processor Performance comparisons.

Table 67. AS/400 Model Comparison Summary - 4 of 4

AS/400 Model	Internal Relative Processor Performance	Memory Range (MB)	Disk Capacity Max (GB)	Comm Lines	LAN Lines	Max Twinax WS	Max ASCII WS	CPU Sec per Tran Guide
9402-2030	2.5	8-24	1.0 - 12.4	0-20	0-2	280	126	<.40
9402-2031	4.0	8-56	1.0 - 12.4	0-20	0-2	280	126	<.30
9402-2032	5.8	16-128	1.0 - 12.37	0-20	0-2	280	126	<.20
9406-2040	4.0	8-72	1.0 - 55.1	1-33	0-4	1000	450	<.25
9406-2041	5.8	16-80	1.0 - 55.1	1-33	0-4	1000	450	<.25
9406-2042	7.2	32-160	1.0 - 55.1	1-33	0-4	1000	450	<.25
9406-2043	10.9	64-832	1.0 - 149	1-64	0-6	2400	1080	<.20
9406-2044	19.0 (2)	64-832	1.0 - 149	1-64	0-6	2400	1080	<.20
9406-2050	23.2	128-1536	1.0 - 260	1-96	0-8	4800	2160	<.15
9406-2051	41.3 (2)	128-1536	1.0 - 260	1-96	0-8	4800	2160	<.15
9406-2052	61.2 (4)	128-1536	1.0 - 260	1-96	0-8	4800	2160	<.15

Table 68. AS/400 Server models compared to B10

type	Model	Memory Capacity (MB)	Disk Capacity (GB)	Interactive Processor Rating	Non-Interactive Processor Rating
9402	100	16-56	1.0-7.9	1.9 = F02	5.9 = F45
	20S/2010	16-128	1.0-12.4	1.9 = F02	5.9 = F45
9404	135	32-384	1.0-27.5	3.3 = F10	10.9 = 1.1 x F50
	140	64-512	1.0-47.2	4.0(2) = F20	22.5 = 1.1 x F70
9406	30S/2411	32-384	1.0-47	3.3 = F10	10.9 = 1.1 x F50
	30S/2412	64-832	1.0-47	4.0(2) = F20	23.5 = 1.2 x F70

Table 69. 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 4, 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. PRTTNSRPT - Job Summary report shows CPU Queuing Multiplier. This

Table 70 (Page 1 of 2). CPU per Transaction Complexity Boundaries

CPU	simple/medium boundary	medium/complex boundary	complex/very complex boundary
100	0.238	0.322	3.218
135	0.137	0.185	1.851
140	0.227	0.307	3.068
B10	0.454	0.614	6.136
B20	0.270	0.365	3.652
B30	0.362	0.490	4.902
B35	0.299	0.405	4.052
B40	0.247	0.334	3.335
B45	0.195	0.263	2.635
B50	0.137	0.185	1.851
B60	0.089	0.120	1.201
B70	0.067	0.090	0.900
C04	0.436	0.590	5.903
C06	0.360	0.487	4.869
C10	0.360	0.487	4.869
C20	0.249	0.337	3.368
C25	0.212	0.287	2.868
D02	0.356	0.482	4.819
D04	0.306	0.414	4.135
D06	0.234	0.317	3.168
D10	0.251	0.340	3.402
D20	0.196	0.265	2.651
D25	0.133	0.180	1.801
D35	0.185	0.241	2.451
D45	0.123	0.167	1.668
D50	0.100	0.135	1.351
D60	0.055	0.075	0.750
D70	0.041	0.055	0.650
D80	0.047	0.063	0.634
E02	0.295	0.399	3.985
E04	0.234	0.317	3.168
E06	0.179	0.242	2.418
E10	0.174	0.235	2.351
E20	0.136	0.183	1.834
E25	0.112	0.152	1.517
E35	0.136	0.183	1.834
E45	0.095	0.128	1.284
E50	0.073	0.098	0.984
E60	0.047	0.063	0.634
E70	0.033	0.045	0.450
E80	0.037	0.050	0.500
E90	0.041	0.055	0.550
E95	0.044	0.060	0.600
F02	0.238	0.322	3.218

Table 70 (Page 2 of 2). CPU per Transaction Complexity Boundaries

CPU	simple/medium boundary	medium/complex boundary	complex/very complex boundary
F04	0.181	0.245	2.451
F06	0.137	0.185	1.851
F10	0.137	0.185	1.851
F20	0.113	0.153	1.534
F25	0.096	0.130	1.301
F35	0.096	0.130	1.301
F45	0.076	0.103	1.034
F50	0.047	0.063	0.634
F60	0.032	0.043	0.434
F70	0.023	0.032	0.317
F80	0.025	0.033	0.334
F90	0.030	0.040	0.400
F95	0.035	0.047	0.467
F97	0.030	0.040	0.400
2010	0.238	0.322	3.218
2030	0.181	0.245	3.451
2031	0.113	0.153	1.534
2032	0.078	0.105	1.051
2040	0.113	0.153	1.534
2041	0.078	0.105	1.051
2042	0.063	0.085	0.850
2043	0.041	0.055	0.550
2044	0.047	0.063	0.634
2050	0.018	0.025	0.250
2051	0.020	0.027	0.267
2052	0.030	0.040	0.400
2114	0.181	0.245	2.451
2115	0.137	0.185	1.851
2411	0.137	0.185	1.851
2412	0.227	0.307	3.068
P02	0.181	0.245	2.451

Note:

This table is updated for V3R1. The Transaction report now uses the hardware file used by BEST/1. The values are equal to or slightly less than previous releases for most system models and system feature numbers.

Table 71. Job Activity and Disk I/O Cross Reference Chart

Job Activity	Physical I/O Type	Logical I/O Type	Fault Type	I/O Counted In:
Read a record randomly	S-DBR	Read	DB	Job
Read a record sequentially	A-DBR	Read		Job
Search access path	S-DBR	Other	DB	Job
Re-page a record	S-DBR		DB	Job
Update a record	A-DBW	Other		Job
Add a record	A-DBW	Write		Job
Delete a record	A-DBW	Other		Job
Update a record (Force Write Rat > 0)	S-DBW	Other		Job
Add a record (FWR > 0)	S-DBW	Write		Job
Delete a record (FWR > 0)	S-DBW	Other		Job
Open a file	S-NDBR		NDB	Job
Close a file	S-DBW			Job
Journaling	S-NDBW			Job
Get a user program	S-NDBR		NDB	Job
Re-page a user program	S-NDBR		NDB	Job
Get a system (VLIC or HLIC) program	S-NDBR			Job
Read the PAG	S-NDBR		NDB if purge is *NO	Job
Re-page the PAG	S-NDBR		NDB	Job
Write the PAG	S-NDBW			Job (PURGE(*YES) only)
Read a data area/data queue	S-NDBR		NDB	Job
Read a subfile	S-NDBR		NDB	Job
Modify a source member using SEU	S-NDBR		NDB	Job
Display messages	S-NDBR		NDB	Job
PCS File Transfer to AS/400	S-NDBR/ S-DBW		NDB	PCS
PCS File Transfer from AS/400	S-NDBR		NDB	PCS
PCS Shared Folders	S-NDBW/ A-NDBW			System

The contents of Table 72 should be viewed as "indicators" that the associated job must be analyzed for the reasonability of the work being done. For example, a job may be updating two sets of similar database data. This could cause the count of physical I/Os to be very high. If this double updating is actually required, then exceeding the database I/O guidelines must be accepted.

Table 72. Disk Physical I/O per Transaction Guidelines

Type of Synchronous I/O	Number of I/Os - 9332/9335/Internal Disk(*)	Number of I/Os - 9336	Number of I/Os - 9337(*)
DB Reads	20 or less	25 or less	25 or less
DB Writes	10 or less	13 or less	13 or less
Total I/O	50 or less	65 or less	65 or less

Note: (*) Depending on your data access patterns, the 9337 DASD read ahead buffers and the 6501, 6530, and 6502 disk controller write cache support may support higher disk I/O rates satisfactorily.

The contents of Table 73 includes average disk service time, internal lab batch benchmark run time and Ops/Sec/GB guideline values that can be used as reference when considering upgrading to newer disk configurations and manually modeling batch job run time based on disk I/O operations.

Use of BEST/1 is still the recommended tool for capacity planning, though you have to do additional calculations for estimating batch job run time. The table is an excerpt from the February 1995 *Performance Capabilites Reference* manual.

Table 73. Disk Service, Batch Run Time, Ops per Second per GB

Disk Model and IOP	GB of Storage per Arm	Typical Service Time	Batch Run Time (Hrs)	Batch Non-RAID Ops/Sec per GB	Batch RAID Ops/Sec per GB
9332 with 6110/6111	.300	31.7	8.0	45	NA
9332 with 6112	.300	29.4	7.7	45	NA
9335 with 6110/6111	.425	26.6	6.8	35	NA
9335 with 6112	.425	25.1	6.6	37	NA
9336-020 with 6112	.875	19.4	5.7	24	NA
9336-025 with 6112	.875	16.7	5.2	28	NA
9337-010 with 6500	.542	19.0	5.6	39	NA
9337-015 with 6500	.542	15.0	5.2	49	NA
9337-020 with 6500	.970	19.0	5.6	22	NA
9337-025 with 6500	.970	16.6	5.2	25	NA
9337-040 with 6500	1.967	17.1	5.2	12	NA
9337-110 with 6500	.407	19.0	7.8	NA	23
9337-115 with 6500	.407	15.0	7.2	NA	30
9337-120 with 6500	.728	19.0	7.8	NA	13
9337-125 with 6500	.728	16.6	7.2	NA	15
9337-140 with 6500	1.475	17.1	7.2	NA	07
9337-210 with 6501	.542	12.1	4.3	51	38
9337-215 with 6501	.542	09.5	4.0	66	49
9337-220 with 6501	.970	12.1	4.3	29	21
9337-225 with 6501	.970	10.7	4.0	33	24
9337-240 with 6501	1.967	11.0	4.0	16	12
9337-420 with 6501	.970	10.0	3.8	38	34
9337-440 with 6501	1.967	08.6	3.5	22	20
9337-480 with 6501	4.194	08.9	3.6	10	09
320MB with 6530	.320	17.7	5.8	71	NA
400MB with 6530	.400	16.6	5.7	60	NA
988MB with 6530	.988	15.6	5.4	26	NA
6602 with 6530	1.031	13.5	4.4	29	NA
6603 with 6530	1.967	14.0	4.4	15	NA
6606 with 6530	1.967	11.5	4.0	18	NA
6607 with 6530	4.194	11.9	4.0	08	NA
320MB with 6502	.320	17.7	5.8	71	NA
400MB with 6502	.400	16.6	5.7	60	NA
988MB with 6502	.988	15.6	5.4	26	NA
6602 with 6502	1.031	10.1	3.9	34	28
6603 with 6502	1.967	10.5	3.9	17	14
6606 with 6502	1.967	08.7	3.6	21	17
6607 with 6502	4.194	09.0	3.6	09	08
2800	.320	19.5	5.9	64	NA
2801	.988	16.8	5.3	24	NA
2802	1.031	13.5	5.0	29	NA

Note:

- NA indicates "not applicable" as under the RAID heading where disk configurations do not support RAID-5.
- The above figures are averages with the disks at approximately 40% busy.

  The above values were derived from a Rochester lab unique batch application and do not include all test results. The *Performance Capabilities Reference* manual contains the complete test results.
- Recall that the 6502 (internal) and 6501 (external IOP) have 2MB and 4MB write caches which enable a significant number of application output operations to be performed per physical I/O operation
- For both the 9337-4xx and the 6606 and 6607 disk, the non-RAID-5 and RAID-5 physical disk I/O guidelines are very close to each other.
- The service times shown are in milliseconds for non-RAID-5 configurations. When RAID-5 is used, add 2 milliseconds to service time for a reasonable approximation.

The older 9332, 9335, and 9336 disks have comparatively high non-RAID-5 I/O operations per second rates. Note, however, their relatively longer service times are clear indicators of poorer performance compared to the 6606/6607, 9337-2xx, and 9337-4xx disks. Application performance with these newer disks and associated I/O processors is approximately 20% to 30% faster than the older 9332, 9335, and 9336 disks.

Appendix B. Field Descriptions and Sample Performance Reports

The *AS/400 Performance Tools Guide*, SC41-8084, contains descriptions of the various printed report fields. The reports are generated from performance data files described in Appendix A of the *AS/400 Work Management Guide*, SC41-8078. Please refer to manuals mentioned above for a discussion on the fields in the Performance reports **before** using this section, because our intention is only to clarify and provide additional information for some of these fields.

Note: Most of the reports use Performance Monitor ″sample″ data, but the Transaction, Lock and Batch Job Trace reports use ″trace″ data. Selecting trace data is an option available when running the STRPFRMON command.

B.1 Job Types

Some jobs, such as QLUS and Licensed Internal Code tasks (for example, storage management tasks, communications tasks, error logging tasks) and PC Support Shared Folder Type 2 tasks (V2R3) and Client Access/400 database and file serving work are not included. in System Summary report output.

For V3R1, System Report 5250 emulation sessions such as RUMBA/400 and PCS5250 are listed as ″Client Access″ under the **Job Type** heading.

B.1.1 Interactive Workload

- Type I - Interactive

  An interactive job is one started by signing on to the system from a display device. Typically, an interactive transaction (represented by pressing the Enter key or a function key) would involve low CPU and disk utilization. Refer to Appendix A, “Guidelines for Interpreting Performance Data” Table 64 on page 346 and Table 72 on page 352.

  An interactive job may perform ″batch-type″ functions that consume relatively high percentages of CPU or perform a relatively high number of database operations. Batch type functions at an ″interactive priority″ such as 20, can adversely affect overall response time and system activity.

- Type C - Client Access/400

  Interactive PC Support functions (Workstation Support Functions - WSF) and Client Access/400 RUMBA and PCS5250 are included in this category.

  These job types are interactive 5250 workstation jobs using PC-based virtual displays. The device names of these jobs are the name of the PC control unit name with suffix Sn. If four 5250 sessions are defined and signed on, four ″interactive PC jobs″ will be identified as *pcctlnam*S1 - *pcctlnam*S4.

  Interactive PC jobs use display station pass-through (DSPT) support, but are identified separately as evoked batch jobs.

- Type D - Target DDM

  Target DDM jobs are started up either by a job on a ″remote system″ or pre V3R1 PC Support Shared Folder Type 0 and 1 functions. There is no way to associate an end user folder function with one of these DDM server jobs even though there is one DDM server job for each PC using Folder Type 0 or 1 support. (PC Support Shared Folder Type 2 jobs do not use DDM jobs and are included under ″batch″ jobs under the report heading of Non-Interactive Workload job types.)

For other V3R1 Client Access/400 jobs (file serving, database serving, etc., see 10.13.2, “Client Access/400 Work Management” on page 292 for more information.

- Type P - Pass-through

  These job types are interactive 5250 workstation jobs using target system DSPT support, separate from the WSF-Client Access/400 support.

  Each display station pass-through job actually involves two OS/400 jobs. One for the virtual display device and the second an APPC job. The associated APPC job consumes very little CPU during the pass-through session. There is no system-provided facility to match the corresponding virtual display job and APPC job together.

- Type T - MRT

  Only an S/36 environment can start up a Multiple Requester Terminal program, which becomes a ″batch″ job that ″takes over″ the workstation while it is connected to the MRT. All transactions from all attached workstations are charged to the MRT job. When the MRT job ″releases″ the workstation, all subsequent transactions from that workstation are charged to the original workstation job.

- System/36 Jobs

  All System/36 environment jobs that are **not MRT jobs** are listed under Interactive Workload jobs even if they ran in batch.

B.1.2 Non-Interactive Workload

In general, non-interactive jobs are not associated with a workstation during processing. However, a ″non-interactive job″ may perform ″interactive-type″ functions regardless of system classification of job type. In the Summary report, the following OS/400 jobs are not included in the report data:

- Subsystem monitors
- QPFRADJ
- QLUS
- QDBSRV1, QDBSRV2
- QSYSARB
- SCPF

The following are the non-interactive job types listed on the System report.

- Type A - Autostart

  Autostart job entries in subsystems initiate autostart job.

- Type B - Batch

  Batch jobs are started using the Submit Job or Start Database Reader command. They also include PC SERVER jobs started by PC Support Shared Folder Type 2 functions. Each PC using Folder Type 2 Support is associated with a server batch job.

- Type E - Evoke

  These jobs are non-DDM jobs started by receiving a valid Program Start Request (Evoke) from a remote system. Prestart jobs are included. The APPC job partner of a display station pass-through virtual workstation job is included here. This is true both for WSF and non-WSF display station pass-through interactive jobs. There is no way to associate the APPC Evoke job with its partner virtual workstation job.

- Type R/W - Spool Reader/Writer

  These are spool reader and writer jobs. They include APF print driver jobs.

B.2 I/O Counts

System data management components increment counters for I/O operations and store the counts for database logical I/O, communications I/O and printer lines/printed pages in the job′s Work Control Block (WCB) which is an internal system object.

The contents of these counters may be displayed when the job is active, using the Display Job (DSPJOB) command. Selecting option 14 (Display Open Files) shows the I/O counts of the currently opened files, regardless of whether the Performance Monitor is running or not. When the job ends or the files are closed, these counters are no longer visible, but have been recorded in the WCB. If the Performance Monitor is running, these counters are recorded in the monitor database files. These I/O counts are totalled for all jobs within a job type as shown on the System Summary report.

B.2.1 Logical Database I/Os

These counters are incremented by one, for every movement of data between the Open Data Path (ODP) buffer and VLIC database buffers.

Examples of database operations that *do not* increment this count include ″no record found″, ″end of file″, ″record lock time-out″ conditions and HLL operations such as Force End of Data and ″position the file″ (RPG SETLL, COBOL START) operations.

Some HLL compliers support their own record blocking and deblocking of buffers exchanged with database management. In these cases a default or OVRDBF SEQONLY(*YES number-of-records) blocking factor is used. For example, assume a program opens a file for input-only processing and the blocking factor is determined to be 14. The first call to database increments the I/O count to 1 but returns 14 records to the RPG program. When the user program issues its 15th READ operation, a second call is made to QDBGETSQ and the I/O count is incremented to 2. If the job ended there, its logical DB I/O count would be 2.

Logical DB I/Os are representative of the amount of work done by the program, but only indirectly affect the actual physical disk reads and writes (both synchronous and asynchronous) performed by the system.

B.2.2 Physical Disk I/Os

These counters are incremented for every transfer of information between the disk device and the LIC buffers. The physical disk I/O value represents the actual workload on the disk actuators. Physical disk I/Os can be caused by database and non-database activity and also result in synchronous or asynchronous workload.

Synchronous and asynchronous I/O counts are the actual transfer of data between a disk and the system. These **physical** I/Os are distinct from logical

I/Os. This I/O can be caused by actual program database I/O operations, system functions that access other non-database objects such as library information, configuration objects, and program objects. Refer to Chapter 3, "Factors Affecting Performance" on page 11 for a details on synchronous and asynchronous disk I/Os.

B.2.3 Communications I/Os

This is the number of user program calls to OS/400 Communications Data Management programs to perform read or write operations on supported ″device types″.

This count is incremented once per call for all ICF file operations other than open and close instructions, and includes all options on write, read and accept input (read-from-invited device) operations.

CPI-C (Common Programming Interface - Communications) I/O functions are also included in this count.

Note: Workstation job I/O operations to display files are not counted as communications I/O operations. Therefore, it does not include remote workstation activity. Communications I/O performed under display station pass-through by the associated APPC job is not counted because the I/O is performed by the LIC.

The communications activity can be viewed interactively with option 17 of the Display Job Menu - ″Display communications status, if active.″

No blocking is performed by the HLL for ICF even though ICF data management may perform blocking based on protocol dependent parameters such as SNA MAXLENRU and BSC BLKLEN parameters.

If the job is using source Distributed Data Management (DDM) functions, its communications I/O counts will be incremented for each HLL program call to a database function by two for each user program function that results in a physical transfer of ″data″ across the communication line. For example, an RPG program CHAIN op code results in a communications PUT to send the ″get-by key″ request to the target DDM job and a communications GET to receive the record.

Once this DDM process has completed, the record is returned to the program and the database logical I/O count is incremented by 1 just as if the record were accessed locally.

B.3 Printer Lines and Pages

This count is incremented by the printer data management modules each time the job causes a print output to be generated. It does not reflect what is actually printed. Spool writer activity is not included in this count but it reflects job logs that are created.

B.4 Disk Utilization

Disk utilization measures the activity of the disk actuators (access arms), and is the most critical factor in determining the impact of disk I/O on performance.

B.4.1 Percentage Busy

Disk utilization is often expressed as a percentage of its capacity to perform work. It is this value that would be used to compare against utilization guidelines.

B.4.2 Operations Per Second

The ″Operations per second″ value in the **Resource Utilization Expansion** section of the System report, indicates the average number of physical disk I/Os performed by each device, and includes checksum and mirroring I/O activity. The synchronous and asynchronous disk I/O values shown in a prior page of the same report, listed by job priority, do not include the additional write activity resulting from mirroring.

Note: Mirroring is indicated on the **Disk Utilization** section of this System Summary report when ″A″ and ″B″ suffixes are shown for disk units (for example, units 0017A and 0017B).

B.5 Disk Percentage Full

This value indicates the amount of disk space occupied by information, and should not be confused with the measurement of how busy the actuators (disk arms) were. The amount of disk space used is not particularly important unless the percentage and amount of available space is very high. Very high disk space occupancy may result in fragmentation of available free space, causing future allocation of disk to occur in extents of less than the maximum 32KB.

B.6 Measured Profile

When printing the System report, entering a name for the ″Profile Name″ prompt (MSRPRF) will generate a measured profile record. This was used with the capacity planner (MDLSYS) in previous releases. This measured profile record may be used for trend analysis, for example, where trend information spans releases of OS/400. However, this profile does not include PC Support/400 Work Station Function (WSF), PC Support/400 File Transfer and Shared Folder, Client Access/400 5250 emulation and server functions, or Target Pass-through workload.

The BEST/1 capacity planner should be used to obtain optimum predictive results with performance data gathered using OS/400 releases from Version 2 Release 2 onwards. This information will include:

- PC Support/400 performance report information
- Client Access/400 performance report information
- Pass-through

which is not available with the measured profiles generated by the System Summary report option, and MDLSYS was not able to manage workload created by these transactions.

Refer to Appendix C, "IBM Internal Use Only Tools/Documents" C.6, "Work Station Function (WSF) Measured Profile" on page 418 for a way to include some of this workload in the Measured Profile. **However, this will impact any subsequent performance models created by BEST/1 on that data set.**

Refer to *AS/400 Performance Capacity Planning V2R2* and *Performance Tools Guide* for more information on capacity planning.

B.7 Active Workstations

This is an estimated value based on the average user key-think time and the average response time. The estimate of active workstations is the number of workstations required to perform the measured number of transactions, assuming all experienced the average response time and active key-think time.

```
Average Response time          = R secs
Average Active Key-Think Time  = K secs
Transactions Rate per hour     = M

Transaction time               = (R + K) secs

                                  3600 secs
Transactions per hour (T)      = ------------ per workstation
                                 (R + K) secs

                                  M
M transactions per hour require = --- workstations
                                  T

                                 M x (R + K)
Estimated Active Workstations  = -----------
                                    3600
```

B.8 Overcommitment Ratio

The Planning Results screen lists the Over-Commitment Ratio (OCR). This over-commitment ratio is the product of the *average working set size* multiplied by the *average number of active workstations* divided by the *total interactive pool size*. As OCR increases above one, page faulting may become a concern.

B.9 Working Set Size

The working set size is derived from the CPU time per transaction with the CPU time for disk I/Os removed. The derivation uses an internal table which is based on actual measured benchmark data where working sets and CPU times were actually known.

Working set size is the amount of job and data storage required for an application to run at peak performance with little or no page faulting.

This is an estimation as no performance monitor data exists to accurately compute the working set size on the AS/400. See index entries for "working set" for additional information.

B.10 Excess Activity Level Time

The Job Summary - System Summary Data report lists a value labelled ″Excs ACTM/Tns″, and is a component of Exceptional Wait time. This value is an average time that a job was assigned an activity level, but did not use the CPU. The *Performance Tools Guide* indicates a value greater than .3 seconds is a signal to investigate what jobs could be contributing to this high value. Typical causes would be jobs at equal to or higher priority consuming excessive CPU, such as programs looping, doing significant mathematical operations, or batch (or ″batch type″ interactive) jobs running at higher priority, or a very large number of equal priority jobs active.

B.11 Program Exceptions

Program exceptions help identify conditions that may result in high resource utilization of system resources, particularly CPU and disk. A very high count of program exceptions could adversely affect performance.

B.11.1 Size

An arithmetic size overflow occurs when the result of an arithmetic operation exceeds its result field capabilities. In applications that are not using data truncation as normal processing, high values of arithmetic overflow not only impact performance but also the validity of data. Depending on the nature of the arithmetic operation causing the overflow, the Binary, Decimal or Floating Point Overflow counters are incremented.

The sum of the binary and decimal overflow counters are listed as ″Size″ exceptions.

B.11.2 Decimal Data

Application program/data cause these exceptions. They occur when data that is not valid is detected by an arithmetic instruction. Examples are sign and digit codes that are not valid for an arithmetic operation.

B.11.3 Authority Lookup

These are really not exceptions. The Authority Lookup count is an indication of additional processing done during object access to implement security.

See commands AUTHTRC and AUTHPRT in IBMLIB.

B.11.4 PAG Faults

Process Access Group (PAG) faults too are not exceptions or error conditions. They are the number of PAG faults that occurred during the period, and excessive counts could highlight potential performance concerns.

B.11.5 Conflicts

The sharing of resources by jobs is controlled by the OS/400 programs and LIC. A conflict occurs when one job has control of an object that is requested by another job. When two jobs attempt to perform mutually exclusive operations on an object, a conflict resolution methodology has to be employed by the system to ensure data integrity. This scheme uses two mechanisms of seizes (in the LIC) and locks (in the OS/400). Seizes and locks are an integral part of normal system operation, and they usually last for only very short periods of time, and ensure system and data integrity.

Seize Conflicts: Seize requests are an unconditional LIC operation which must complete. That is, it does not timeout if it is unable to complete before a particular time. Any job waiting on a seize stays in the activity level because (1) it is in the process of executing an MI instruction, and (b) seizes are assumed to be of very short duration (usually a few milliseconds). An example of a seize is when a record is updated that involves maintenance of access paths. The entire physical file and and all associated logical files will be seized until the update is complete. This ensures integrity of data before users are allowed access to the database. The greater the number of immediately maintained access paths, the more likely that seize conflict will occur.

Lock Conflicts: Lock conflicts are conditional and would timeout after a predetermined period of time. Lock waits cause the job to lose its activity level until the request can be completed. Object locks can result from explicit commands (ALCOBJ) or database record locks may occur when a record is read for update.

B.11.6 Verify Exceptions

Verify exceptions occur when a pointer is not resolved (for example, when another program is first called from within a job using a variable), a referenced object doesn′t exist (as in a test for object existence using the CHKOBJ command) or when a ″domain″ violation occurs (executing an MI instruction that is blocked, other than under level 40 security).

B.11.7 EAO Exceptions

An Effective Address Overflow Exception occurs when machine instruction ″base register and displacement addressing″ exceeds a currently implemented limit. The Machine Interface (MI) instructions used by the operating system are unaware of any implementation restrictions. So whenever an address limit is reached, the machine processes the exception and the MI instruction completes successfully. These exceptions consume very little processing time. See Appendix A for their impact based on CPU model. If a large percentage of CPU can be attributed to EAOs, the problem must be reported to IBM service. Refer to Appendix A, "Guidelines for Interpreting Performance Data" for guidelines on the effect of excessive exception counts on CPU utilization.

An application design could be the cause when the number of EAO exceptions per second takes up significant CPU utilization. One cause could be the use of a very large (for example, 1MB of storage) user data space or an array where there is repetitive processing that addresses both the beginning and ending addresses of the space or array.

B.12 Permanent Writes

A permanent write is any write to a disk that changes a permanent object, versus a temporary object. This could include file add, update, delete functions, clearing objects, and updating objects. Examples of temporary objects are job work areas, objects in QTEMP libraries etc.

B.13 Disk Buffer Overrun/Underrun

B.13.1 Overrun

This is a count of the number of times that data was available to be read into the disk controller, but the disk controller buffer still contained valid data that was not retrieved by the storage controller. Consequently, the disk had to wait until the buffer was available to accept data.

B.13.2 Underrun

This is a count of the number of times that the disk controller was ready to transfer data to the disk on a write, but the disk controller buffer was empty. The data was not transferred in time by the disk IOP to the disk controller buffer. Consequently, the disk had to wait until the buffer contained data.

B.14 OPSTART Msg

This is a count of the messages sent on the I/O bus between the system and the IOP that is used to initiate work in either the system or the IOP. This count provides a relative measure of performance between the system and all IOPs on the system.

B.14.1 Reverse

A Reverse OPSTART count is the number of messages flowing from the IOP to the system. For communications, these are typically the result of inbound data from all the lines attached to the IOP, or from the local workstations. A significant number of these messages is not expected for storage media IOPs. While the normal function of Reverse Opstart messages is for inbound data, they are also used for non-steady state functions like IOP initialization and notification of IOP detected errors.

B.14.2 Normal

This counts the messages flowing from the system to the IOPs. They are the result of outbound data from the system and destined for the lines attached to the IOP, to the local workstations or storage media.

B.15 Bytes Transmitted

This value measures the number of bytes transmitted across the I/O bus between the system and IOP. The count includes communications data as well as control structures for managing the data.

B.15.1 IOP

This is the number of bytes transmitted from the IOP to the system, and includes inbound data from all the lines on this IOP.

B.15.2 System

This is the number of bytes transmitted from the system to the IOP, and includes outbound data to all the lines on this IOP.

B.16 Restart Queue

The restart queue is a count of the number of times that the IOP sent a work request (OPSTART) to the system, and the system queues to process the requests were full. This value should normally be zero. It is an indicator that the IOP is sending request and data to the system faster than the system can handle them.

B.17 BNA Received

Buffer Not Available is a count of the number of times that the system hardware buffer overrun occurred. That is, the IOP work requests were being sent to the system faster than the hardware buffers could be unloaded. This value should typically be zero.

B.18 Transaction Complexity Classification

The Performance Tools Transaction Report - Job Summary lists the interactive transactions in categories of complexity. Refer to Appendix A, "Guidelines for Interpreting Performance Data" for the CPU utilization boundaries used in the classification of transactions.

B.19 Batch Thread

A Batch Thread represents a sequence of single batch jobs running at a particular job priority. If multiple batch jobs execute concurrently, each of them would be assigned to seperate batch threads. For example, if two jobs run sequentially they will show up as two jobs in the same thread. If two jobs run concurrently, they will show up as being in two different threads. Therefore, the total number of batch threads listed in the Batch Thread Analysis Report will indicate the maximum number of concurrently executing batch jobs during the period trace data was collected. The batch threads are listed in sequence by priority.

B.20 BMPL

The values refer to the number of other jobs at the **B**eginning of a transaction which occupied a **M**icro**P**rogramming **L**evel (Activity Level) in the storage pool in which the transaction occurred. The number of jobs in the activity level are presented in the Transition report as either (a) currently in an activity level or (b) in the ineligible queue.

B.21 Transaction Boundary Wait Codes (Trace Record)

A Transaction is a basic unit of work done on a system. In the case of an interactive transaction, it represents the period between the user making a request of the system by pressing the Enter key or a function key, until the system responds to the user′s request.

The progression of a transaction through the system includes transitions in state, which are reported when a Transition report is selected when running the Transaction Report (PRTTNSRPT) command.

B.21.1 EOTn

End of the transaction.

B.21.2 EORn

End of response time for the transaction.

B.21.3 SOTn

Start of a transaction.

In the above trace codes, ″n″ is a numeric variable that represents the type of transaction involved:

- 1 = Display I/O
- 2 = Data Queue
- 3 = MRT
- 4 = Source Pass-Through
- 5 = Target Pass-Through
- 6 = WSF Target Pass-Through

See Figure 87 on page 394 for a sample transaction report-transition detail report showing transaction boundaries.

Also refer to Appendix D, “Performance Tools/400 Transaction Boundary Overview” on page 439 for more detail.

Refer to the QTRDMPT file format in *AS/400 Performance Tools Guide* for more details.

B.22 System, Component and Transaction Report Differences

In some areas discrepancies have been identified for apparently equivalent data items. Subsequent topics in this appendix discuss some of these differences. In general, these differences can be attributed to the difference between analyzing Performance Monitor *sample* and *trace* data as follows:

- System Report (sample data)
- Component Report (sample data)
- Transaction Report (trace data)
- Lock Report (trace data)
- Job Report (sample data)
- Pool Report (sample data)

- Resource Report (sample data)
- Batch Job Trace Report (trace data)

In most cases where there is a discrepancy, reports based on trace data contain the more accurate data. Some reports include all interactive work and others include only ″non-programmable display device″ jobs under that report′s interactive data. All known differences are not included in this publication.

Refer to Appendix B of the *AS/400 Performance Tools Guide* for a discussion on the transaction boundaries used by the different commands used in performance analysis.

B.22.1 Response Time Differences

The following are known differences between reports showing interactive transaction average response time.

- If Disconnect Job is used with an interactive workstation the time while disconnected in not included in the Component report response time, but is included in the Transaction report response time. Use the Component report response time in your analysis.

  The Transaction report-Job Summary report, System Summary Data *does not include WSF jobs for interactive job type average response time*.

B.22.2 Multiple Requester Terminal (MRT) Transactions

On the System report, System/36 MRT transactions are categorized as Interactive Work Load transaction information. The batch job started by the MRT request is reported as a batch job on the Component and Transaction reports.

B.23 Component and Job Report Differences

B.23.1 Batch Job Elapsed Time

The elapsed job time shown under the Non-interactive Job Detail section of the Print Job Report (PRTJOBRPT) command and Elapsed Seconds on the Print Transaction Report - Job Summary, Batch Job Analysis section include the entire interval the job started in if the job was started while the Performance Monitor was collecting data. If long sampling intervals were used, this could indicate the job was active much longer than it actually was.

The Print Transaction report elapsed seconds will be made accurate in the next release.

However, this longer time can skew the CPU utilization shown for the job in the reports. The CPU utilization is accurately shown for non-interactive jobs on the Batch Job Trace output for the Print Trace Report (PRTTRCRPT) command.

Note that the PRTTNSRPT OUTPUT(*FILE) command produces files QTRJSUM and QTRJOBT. The PRTJOBRPT command processes file QTRJOBT. Field JELAP in file QTRJSUM contains the actual job elapsed time as Performance Monitor trace data is used.

B.24 Charging Resource Utilization to Interactive Programs

Use the Transaction and Transition reports to do this analysis.

The Job Summary Report - Individual Transaction Statistics and Interactive Program Statistics data list several categories of performance metrics that are commonly referred to as the ″ten worst.″ Some of these metrics are the ″transactions with the longest short wait time,″ ″transactions with the longest lock wait time,″ ″transactions with the longest CPU service,″ and so on. Each of these categories identify a program that is charged with consuming that resource.

The program name listed is the program that first does a workstation output operation following the receipt of workstation input (which signalled the beginning of a transaction to the system). In many application environments this accurately reflects the program doing the work. However, in many other application environments, programs *called by that program* actually consumed the system resource. Therefore, further analysis of the identified program and programs it called is required to accurately assess what program is responsible.

A good example of this situation is the typical inclusion of an OS/400 User Interface Manager program in the list of ″ten worst.″ Frequently programs QUIINMGR and QUIMNDRV are in this list. These programs are almost never responsible for high consumption of a system resource. Rather, it is the functions (and programs called) invoked from a menu screen that are the resource consumer. You need to identify some specific jobs and print a Transition Detail report to identify what is going on below the QUIxxxx program. In some cases, the Timing and Paging Statistics (TPST) PRPQ or the ″as is″ System/36 Environment Trace Job (TRACE36E) tool from VM are helpful in this analysis.

One example of charging program QUIINMGR with the ″highest response time″ can be demonstrated by doing a Send Network File command from a system menu. The SNDNETF function does no workstation I/O and can lock the workstation up while copying a large file to an internal space for delivery later by SNADS support. When SNDNETF completes its work, a message is issued to the operator indicating the results. This message is written by QUIINMGR.

B.25 Licensed Internal Code Tasks

When reviewing the Performance Tools/400 Component Report, BEST/1, or the Work with System Activity (WRKSYSACT) command output, Licensed Internal Code (LIC) tasks will appear. Normally, these tasks take less than .2 percent CPU utilization. In some cases a LIC task will take more than this CPU utilization.

In most performance problem situations, a LIC task is not part of the performance problem. However, any LIC task that approaches 1% CPU utilization needs to be investigated.

Chapter 26 of the *Diagnostic Aids - Volume 1, for Version 3*, LY44-3900, lists many LIC tasks and gives a brief description of their function. This redbook provides LIC task information for some of the tasks that are not included in LY44-3900 for V3R1.

Table 74. Licensed Internal Code Tasks (subset)	
Task Name	**Task Description**
BAITnn	Performs asynchronous writes of changed pages to disk and reads of data, including pre-brings, into main storage for new file systems in the Integrated File System support, for V3R1 QDLS folders, and for V3R1 network files. A new "byte stream file" object is used to hold this kind of data. There may be several ("nn") of these tasks, depending on system activity.
SMTUNE	Storage Management Tuner. This is activated frequently and is the primary task for *expert cache* processing. It will be activated even if expert cache is not activated for a storage pool (*CALC or *USRDFN). If expert cache is not activated for any pools, the task does minimal processing before going into a wait state.
IOPMGR	Inter-Process Communication Facility (IPCF) IOP Manager task
RROUTE	IPCF Resident Router
IOBER	IPCF Bus Error Task
CCBN	IPCF Communications Control Bus Functions Task
R##ccn	IPCF Special Purpose Multiple Routers Task. For all releases previous to V3R1, there is one of these per IOP on a bus. For V3R1 some IOPs have multiple router tasks. For example a LAN I/O Manager (IOM) may have multiple router tasks to improve performance. In this case you may see R251D1, R251D2, R251D3. See table notes for an explanation of ##, cc, n.

Note:

- An IPCF task provides the communication link between mainline LIC code and I/O Processor (IOP) code across the bus.
- R##ccn explanation

```
o ## is the hexadecimal representation of the one byte logical
  bus address of the IOP:

   Bits           Represent
    0-2            Bus number (bus 000 through 010 are valid)

    3-7            Position of the IOP on the bus (bits 00000
                   through 11111 are valid)

o cc is the hexadecimal conversion of part of connection ID. The
  connection id is assigned to each IOP during IPL.

o n is the number of the router (0-9,A-Z for first 36
  routers, # otherwise)
```

B.26 Performance Trace Database Files

When *FILE is specified on the RPTTYPE parameter of the PRTTNSRPT command, corresponding members are created in the following files:

- QTRTSUM

  This file contains one record for every transaction identified by the PRTTNSRPT command. The summary data represents the activity for the transaction.
- QTRJSUM

  The job summary file contains one record for each job or task listed on the PRTTRNRPT job summary report.
- QTRJOBT

  This file contains one record per time slice end for all jobs. Time slice end records are created if the job CPU usage reaches one of the following values:

– External CPU time slice value
– An internal time slice value defined by the STRPFRMON command.

The summary data represents the activity since the last TSE or other multiprogramming level trace record.

If *TRCDTA is specified on the RPTTYPE parameter of the PRTTNSRPT command, a member is created in the QTRDMPT. This file is a version of QAPMDMPT file formatted as a database, giving the user access to performance monitor trace records.

Refer to the *Performance Tools Guide* for details.

B.27 Sample System Reports

Selected report sections show unique V3R1 examples. Other sections with no new V3R1 information remain unchanged.

Workload Section: Interactive Workload – Sample:

```
                                                          System Report                                              3/22/95  9:49:08
                                                            Workload                                                        Page 0001
                                                Remote SQL, ODBC: 2 WIN 3.1 Clients
Member  . . . : RODBCSQL    Model/Serial  . : F25     /10-43592  Main storage . . :   48.0 M   Started  . . . . : 03/21/95 13:47:59
  Library . . : COOK        System name . . :         RCHASM03  Version/Release  :   3/ 1.0   Stopped  . . . . : 03/21/95 14:26:47
Interactive Workload
    Job               Number          Average         Logical DB         ------ Printer -------        Communications         MRT
    Type           Transactions       Response        I/O Count              Lines       Pages           I/O Count          Max Time
-------------      ------------       --------       -----------        ----------     -------       --------------       --------
DDM Server                    1            .00                 0                26           1                    0
Client Access                15           2.06               117               184           6                  191
PassThru                     15           1.13                 6                 0           0                    0
Total/Average                31           1.60               123               210           7                  191
-------------------------------------------------------------------------------------------------------------------------------------
V3R example
```

Figure 43. Workload Section: Interactive Workload

Workload Section: Noninteractive Workload – Sample:

```
Noninteractive Workload
   Job        Number          Logical DB        ------ Printer ------          Communications          CPU per          Logical
   Type       of Jobs         I/O Count            Lines        Pages            I/O Count           Logical I/O      I/O /Second
----------   ---------       -----------       ----------    -------         --------------         -----------      -----------
Batch                1             1,227                0          0                      0                 .01               .6
Spool                1                 0                0          0                      0                 .00               .0
Evoke               25                 0                0          0                      0                 .00               .0
Total/              27             1,227                0          0                      0                 .01               .6
Average
Total CPU Utilization . . . . . . :  41.9
```

Figure 44. Workload Section: Noninteractive Workload

Resource Utilization (First Part) – Sample:

```
                                                          System Report                                              3/22/95  9:49:08
                                                       Resource Utilization                                                 Page 0002
                                                Remote SQL, ODBC: 2 WIN 3.1 Clients
Member  . . . : RODBCSQL    Model/Serial  . : F25     /10-43592  Main storage . . :   48.0 M   Started  . . . . : 03/21/95 13:47:59
  Library . . : COOK        System name . . :         RCHASM03  Version/Release  :   3/ 1.0   Stopped  . . . . : 03/21/95 14:26:47
                    -------------------- Average Per Transaction --------------------
                                                   Sync        Async
     Job            Response         CPU           Disk         Disk           DB
     Type           Seconds        Seconds         I/O          I/O           I/O
--------------      --------       -------        -----        -----        -------
DDM Server                .0            .71        119.0         14.0            .0
Client Access            2.0           2.03        237.0         62.8           7.8
PassThru                 1.1            .51         34.9          1.2            .4
Total/Average            1.6           1.26        135.4         31.4           3.9
-------------------------------------------------------------------------------------------------------------------------------------
V3R1 example
```

Figure 45. Resource Utilization

Resource Utilization (Second Part) – Sample:

```
                                Tns          Active        -------------------------- Disk I/O Per Second --------------------------
      Job          CPU         /Hour        Jobs Per         Total      --------- Synchronous --------     -------- Asynchronous --------
      Type         Util         Rate        Interval          I/O        DBR      DBW      NDBR     NDBW      DBR      DBW     NDBR     NDBW
--------------     ----      ---------      --------       -------     -----    -----     -----    -----    -----    -----    -----    -----
DDM Server           .0             1              0            .0        .0       .0        .0       .0       .0       .0       .0       .0
Client Access       1.3            23             19           1.9        .1       .0       1.2       .0       .0       .0       .0       .2
PassThru             .3            23              1            .2        .0       .0        .2       .0       .0       .0       .0       .0
Total/Average       1.6            48             20           2.2        .2       .0       1.5       .0       .0       .0       .0       .2
V3R1 example
```

Figure 46. Resource Utilization (Second Part)

Resource Utilization Expansion (First Part) – Sample:

```
                                                  System Report                                          3/22/95  9:49:08
                                         Resource Utilization Expansion                                         Page 0003
                                       Remote SQL, ODBC: 2 WIN 3.1 Clients
Member  . . . : RODBCSQL    Model/Serial  . : F25     /10-43592  Main storage . . :   48.0 M   Started  . . . . : 03/21/95 13:47:59
  Library . . : COOK        System name . . :          RCHASM03  Version/Release  :   3/ 1.0   Stopped  . . . . : 03/21/95 14:26:47
                ---------------------------------------------  Average Per Transaction ---------------------------------------------
                -------------------- Physical Disk I/O -----------------------    ------- Logical --------    -- Communications --
     Job        ------- Synchronous --------     ------- Asynchronous --------    -----Data Base I/O -----            I/O
     Type        DBR     DBW     NDBR    NDBW     DBR     DBW     NDBR    NDBW     Read     Write     Other     Get          Put
--------------  -----   -----   -----   -----    -----   -----   -----   -----    -----    -----     -----     -----        -----
DDM Server        9.0     1.0    98.0    11.0       .0     2.0      .0    12.0       .0       .0        .0        .0           .0
Client Access    29.4      .9   196.4    10.2     13.8      .8    13.9    34.2      7.8       .0        .0       3.5          9.2
PassThru          1.0      .0    33.8      .0       .0      .0      .0     1.2       .4       .0        .0        .0           .0
Total/Average    15.0      .4   114.6     5.3      6.6      .4     6.7    17.5      3.9       .0        .0       1.7          4.4
-----------------------------------------------------------------------------------------------------------------------------------
V3R1 example
```

Figure 47. Resource Utilization Expansion

Resource Utilization Expansion (Second Part) – Sample:

```
                Job            CPU        Cum       ------- Disk I/O -------     ----- CPU Per I/O ------     -- DIO /Sec --
Priority        Type           Util       Util         Sync          Async          Sync          Async        Sync   Async
--------    --------------     ----      -----      ----------    ----------     ----------    ----------     -----   -----
  000       Batch               2.4        2.4           9,190         2,238          .0061         .0251       3.9      .9
            System              2.2        4.7          11,126           937          .0047         .0564       4.8      .4
  009       System               .0        4.7               0             0          .0000         .0000        .0      .0
  010       Batch                .0        4.7               7             0          .0035         .0000        .0      .0
  016       System               .0        4.7              15             0          .0020         .0000        .0      .0
  020       DDM Server           .0        4.7             119            14          .0060         .0513        .0      .0
            Client Access       1.3        6.0           3,556           942          .0085         .0324       1.5      .4
            PassThru             .3        6.4             524            18          .0148         .4331        .2      .0
            Batch                .6        7.0           2,685             0          .0052         .0000       1.1      .0
            AutoStart            .0        7.0               2             0          .0050         .0000        .0      .0
            Evoke                .0        7.0             169             8          .0036         .0770        .0      .0
            System               .0        7.0              52             0          .0025         .0000        .0      .0
   .
   .
  060       System               .0       27.1               0             0          .0000         .0000        .0      .0
Total/Average                                           40,800         8,996                                     17.6     3.8
V3R1 example
```

Figure 48. Resource Utilization Expansion (Second Part)

Storage Pool Utilization – Sample:

```
                                                  System Report                                         11/24/90 10:28:19
                                            Storage Pool Utilization                                            Page 0004
                                              Sample System Report

Member  . . . : TEMP         Model/Serial  . . :  B40/10-15005   Main storage . . :   40.0 M   Started  . . . . : 11/02/90 10:01:54
  Library . . : QPFRDATA     System name . . . :  ABCSYSTEM      Version/Release  :   2/ 2.0   Stopped  . . . . : 11/02/90 10:31:51
                                                                     --------- Avg per Second --------     ---- Avg per Minute ----

Pool        Size     Act       CPU       Number        Average        ------ DB -----    ---- Non-DB ---     Act-     Wait-     Act-
 ID         (K)      Lvl       Util       Tns          Response       Fault     Pages    Fault     Pages     Wait     Inel      Inel
----     -------     ---      -----     -------       --------        -----     -----    -----     -----    ------   ------    ------
 01        6,000      00       10.0           0             .0           .0        .0       .0        .1         0        0         0
 02       34,960      25       31.9       3,085             .6          5.2      23.6       .3        .4       104        0         0
Total/    40,960               41.9       3,085             .6          5.2      23.6       .3        .5       104        0         0
Average
```

Figure 49. Storage Pool Utilization

Disk Utilization – Sample:

```
                                                       System Report                                              11/24/90 16:01:39
                                                      Disk Utilization                                                    Page 0005
                                                    Sample System Report

Member  . . . : TEMP        Model/Serial  . . :  B40/10-15005   Main storage . . :   40.0 M   Started  . . . . : 11/02/90 10:01:54
  Library . . : QPFRDATA     System name . . . :  ABCSYSTEM      Version/Release  :   2/ 2.0   Stopped  . . . . : 11/02/90 10:31:51

                   Size     IOP     IOP     ASP     CSS     --Percent--        Op per    K per       ----- Average Time per I/O -----
Unit     Type      (M)      Util    ID      ID      ID      Full      Util      Second     I/O          Service      Wait       Response
----     ----     ------    ----    ----    ---     ---     ----      ----    ---------  --------      -------     ------      --------
0001     9335       427     10.0    0-01    01      00      74.0        .1          .06        .8         .015        .005        .020
0002     9335       427     10.0    0-01    01      00      63.7        .1          .02       1.2         .043        .002        .045
0003     9335       427     15.0    0-02    01      00      63.9        .0          .00       1.7         .000        .000        .000
0004     9335       427     15.0    0-02    01      00      64.0        .0          .00       2.1         .000        .000        .000
0005     9335       427     10.0    0-03    01      00      63.9        .0          .01       2.3         .000        .000        .000
0006     9335       427     10.0    0-03    01      00      63.9        .0          .01        .8         .000        .000        .000
0007     9335       427     15.0    0-04    01      00      64.0        .0          .03        .8         .000        .000        .000
0008     9335       427     15.0    0-04    01      00      63.7        .0          .04        .8         .000        .000        .000
Average                                                     65.2        .0          .21       1.0         .000        .000        .000
```

Figure 50. Disk Utilization

Communications Summary – Sample:

```
                                                       System Report                                              11/24/90 10:28:19
                                                   Communication Summary                                                  Page 0006
                                                   Sample System Report

Member  . . . : TEMP        Model/Serial  . . :  B40/10-15005   Main storage . . :   40.0 M   Started  . . . . : 11/02/90 10:01:54
  Library . . : QPFRDATA     System name . . . :  ABCSYSTEM      Version/Release  :   2/ 2.0   Stopped  . . . . : 11/02/90 10:31:51

                                 Line       Avg  Max    Active            Number                               ------ Bytes per Second -----
BUS/IOP/              Protocol   Speed      Util Util   Devices        Transactions        Response        Received          Transmitted
  Line                --------  -------     ---- ----   --------       -------------       ---------       ---------         -----------
BUS 0  IOP 06 (6110)
  L01ES                 SDLC        9.6       21  65       0                   0                 .0           206.8                 54.4
  L02ES                 SDLC        9.6       22  64       0                   0                 .0           212.0                 54.5
  L03ES                 SDLC        9.6       21  51       0                   0                 .0           207.9                 52.7
BUS 0  IOP 11 (6110)
  RALIN21               SDLC        9.6       44  87       2                   0                 .0           109.4                426.7
  RALIN22               SDLC        9.6       39  72       1                   0                 .0           104.2                375.4
BUS 2  IOP 02 (6110)
  VMBRIDGE              BSC         9.6        2  44       0                   0                 .0            31.1                   .2
```

Figure 51. Communications Summary

Report Selection Criteria: Select Parameters – Sample

```
                         Report Selection Criteria
Select Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting           Payroll              Research
                              Development          ProjectX             MrNolansStaff

                            - No Select parameters were chosen.
```

Figure 52. Report Selection Criteria: Select Parameters

Report Selection Criteria: Omit Parameters – Sample:

```
Omit Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting          Payroll              Research
                              Development         ProjectX             MrNolansStaff

                            - No Omit parameters were chosen.
```

Figure 53. Report Selection Criteria: Omit Parameters

B.28 Sample Component Reports

Selected report sections show unique V3R1 examples. Other sections with no new V3R1 information remain unchanged.

Component Interval Activity – Sample

Component Report 04/20/90 10:06:31
Component Interval Activity Page 0002
Sample Component Report

Member . . . : GOODSTUF Model/Serial . . : B60/10-12883 Main Storage . . : 96.0 M Started . . . . : 3/29/90 11:35:54
Library . . : QPFRDATA System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 3/29/90 13:35:39

Itv End	Tns /Hour	Rsp /Tns	---- CPU Utilization --- Total	Inter	Batch	---- Disk I/O ---- --- Per Second --- Sync	Async	High -Utilization- Disk	Unit	Pool --- Faults/Sec -- Mch	User	ID	Excp
11:40	1,282	.0	26.0	24.7	1.3	4.9	.0	2	0003	0	4	03	1,401
11:45	1,216	.0	7.7	5.8	1.9	16.7	.0	2	0016	0	2	03	2,598
11:50	903	.1	19.5	18.6	.9	4.8	.0	5	0020	0	10	03	520
11:55	794	.2	2.8	2.5	.3	1.3	.0	2	0023	0	1	03	82

Figure 54. Component Interval Activity

Job Workload Activity – Sample:

Component Report 3/22/95 9:49:32
Job Workload Activity Page 2
Remote SQL,ODBC: 2 WIN 3.1 Clients
Member . . . : RODBCSQL Model/Serial . : F25 /10-43592 Main storage . . : 48.0 MB Started . . . . : 03/21/95 13:47:59
Library . . : COOK System name . . : RCHASM03 Version/Release : 3/ 1.0 Stopped . . . . : 03/21/95 14:26:47

Job Name	User Name	Job Number	Typ	Pl	Pty	CPU Util	Tns	Tns /Hour	Rsp	---------- Disk I/O --------- Sync	Async	Logical	Cmn I/O	EAO Excp	PAG Fault	Arith Ovrflw	Perm Write
#-0100			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
#A0003			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
#B000C			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
#B000D			L	01	00	.0	0	0	.00	118	0	0	0	0	0	0	0
#B0011			L	01	00	.0	0	0	.00	7	0	0	0	0	0	0	0
#B0015			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
#B0016			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
. . .																	
. . .																	
R020B0			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
R040E0			L	01	00	.0	0	0	.00	38	0	0	0	0	0	0	0
R040E1			L	01	00	.0	0	0	.00	21	0	0	0	0	0	0	0
R040E2			L	01	00	.0	0	0	.00	9	0	0	0	0	0	0	0
R040F0			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
SCPF	QSYS	000000	X	02	40	.0	0	0	.00	169	35	148	0	0	35	0	15
SFD			L	01	00	.0	0	0	.00	1	0	0	0	0	0	0	0
SFS3			L	01	00	.0	0	0	.00	0	0	0	0	0	0	0	0
SFTR			L	02	00	.1	0	0	.00	122	0	0	0	1	0	0	0
SMAI01			L	02	00	.0	0	0	.00	0	0	0	0	0	0	0	0
SMAI02			L	02	00	.0	0	0	.00	0	0	0	0	0	0	0	0
SMAI03			L	04	00	.0	0	0	.00	0	0	0	0	0	0	0	177
SMAI04			L	02	00	.0	0	0	.00	17	0	0	0	0	0	0	464
. . .																	
. . .																	
TCPIPLOC	QGATE	023425	B	02	40	.0	0	0	.00	8	0	0	0	0	0	0	0
TEAM03	MAATTA	023840	D	02	20	17.9	0	0	.00	119	14	0	0	0	14	0	12
TEAM03	MAATTA	023841	C	02	20	19.6	0	0	.00	96	11	0	0	0	11	0	16
TEAM03	MAATTA	023842	C	02	20	1.7	0	0	.00	520	205	35	42	15	48	0	18
TEAM03	QSYS	023839	E	02	20	20.5	0	0	.00	169	8	0	2	0	5	0	7
TRECBF			L	01	00	.0	0	0	.00	93	0	0	0	0	0	0	91
TRS			L	01	00	.0	0	0	.00	0	26	0	0	0	0	0	0
VLTASK			L	01	00	.0	0	0	.00	3	0	0	0	0	0	0	3
Total/Average						27.1	30	46	1.60	40800	8996	6377	193	741	2666	0	15765

V3R1 example

Figure 55. Job Workload Activity

Storage Pool Activity – Sample:

```
                                                         Component Report                                        04/20/90 10:06:31
                                                      Storage Pool Activity                                              Page 0005
                                                     Sample Component Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :     96.0 M  Started . . . . :   3/29/90 11:35:54
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :     2/ 2.0  Stopped . . . . :   3/29/90 13:35:39

Pool identifier . . . : 01

           Pool                               Avg                ---------- Avg per Second -----------    ----- Avg per Minute -----
 Itv       Size      Act       Total          Rsp        CPU       ------ DB ------    ----- Non-DB -----     Act-       Wait-       Act-
 End       (KB)      Level       Tns          Time       Util      Faults     Pages     Faults     Pages      Wait       Inel        Inel
-----     ------     -----      ------        ----       ----      ------     -----     ------     -----     -----      -----       -----
11:40       3000         7           0          .0        2.8          .0         0         .2         1         5          0           0
11:45       3000         7          32         1.4        2.8          .0         0         .0         0         6          0           0
11:50       3000         7           9          .3        4.5          .0         0         .2         1        14          0           0
11:55       3000         7          17          .5        2.1          .0         0         .0         0         7          0           0
```

Figure 56. Storage Pool Activity

Disk Activity – Sample:

```
                                           Component Report                                04/20/90 10:06:31
                                             Disk Activity                                         Page 0009
                                         Sample Component Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :    96.0 MB  Started . . . . :  10/31/90 09:56:12
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :    2/ 2.0   Stopped . . . . :  10/31/90 11:28:55

                                            --------- Average Disk Activity Per Hour --------
                Srv     ---- Buffer ---     ------------ Disk Arm Seek Distance -------------     --------- Disk Capacity ---------
Unit     Util   Time    Over      Under            0      1/12      1/6     1/3     2/3    >2/3     Permanent       MB        Percent
----     ----   ----    -----     -----     -------  -------   -------  ------  ------  ------     ---------     -------    -------
0001      2.9   .027       0          0         226      782       174     730     353      24             0          13        6.5
0002      2.7   .023       0          1         286      660       239     312     224      52             0          91       45.5
0003      2.4   .024       0          1         243      772       333      99     335      18             0          90       45.0
0004      1.5   .028       0          0         118      454       237      66     351      25             0          91       45.5
0005      2.5   .026       0          0         210      567       294     150     467      32             0          91       45.5
0006      1.9   .026       1          1         149      462       244     201     335      49             0          91       45.5
                                                                     .
                                                                     .
                                                                     .
                                                                     .
Total/    2.3   .025       1          3       1,235    3,699     1,524   1,562   2,067     202             0         467       38.9
Average
```

Figure 57. Disk Activity

IOP Utilizations – Sample:

```
                                           Component Report                                04/20/90 10:06:31
                                           IOP Utilizations                                        Page 0011
                                         Sample Component Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :    96.0 M  Started . . . . :    3/29/90 11:35:54
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :    2/ 2.0  Stopped . . . . :    3/29/90 13:35:39

Communications                              --- OPSTART Msg ---   --- Bytes Transmitted ---    Restart      BNA          Available
IOP′s                    Utilization        Reverse      Normal      IOP           System       Queues      Received      Storage
--------------           -----------        -------      ------   -----------   -----------    -------     --------      ---------
BUS 0  IOP 05                2.0                  0         573       124,141       575,176          0            0      3,835,536
BUS 1  IOP 02                1.0                  0          24             0         4,224          0            0      9,999,999
BUS 2  IOP 02                2.0                  0       1,142       771,556       345,574          0            0      9,999,999

DASD IOPs                Utilization          Ops per Sec
-------------            -----------          -----------
BUS 0  IOP 01                1.0                        1
BUS 0  IOP 09                2.0                        2
BUS 1  IOP 01                1.0                        0

Multi-function
IOPs                     Utilization
--------------           -----------
BUS 2  IOP 01                2.0

Local Work            Controller   -- OPSTART Msg --  Bytes Transmitted  Restart     BNA      ----- Queue Average -----  Twinaxial
Station IOPs    Util     Name      Reverse    Normal    IOP     System    Queues    Received    Wait    Suspend   Active      Util
--------------  ----  ----------   -------    ------  -------  --------  -------    --------   ------   -------   ------  ---------
BUS 0  IOP 03   15.0  CTL01              0      2597   396727   1853723        0           0      7.0      95.0      5.0        7.0
BUS 0  IOP 04   14.0  CTL02              0      2255   790955   2690092        0           0     34.0      64.0     12.0       33.0
```

Figure 58. IOP Utilizations

Local Work Stations – Response Time Buckets – Sample:

```
                                            Component Report                                          1/10/92  6:46:18
                                  Local Work Stations - Response Time Buckets                           Page   21
                                           Sample Component Report

Member  . . . : GOODSTUF    Model/Serial  . . :  B70/10-15018   Main storage . . :   96.0 MB  Started . . . . :  11/08/91 14:16:58
  Library . . : QPFRDATA    System name . . . :      RCHASLA6   Version/Release  :    2/ 2.0  Stopped . . . . :  11/08/91 16:31:41

   Ctl/Device      Util     Bus    IOP
---------------    ----     ---    ---
CTL01               8.0      0      3
                                           0- 1.0      1.0- 2.0     2.0- 4.0     4.0- 8.0     > 8.0     Rsp Time
                                           ------     ---------    ---------    ---------    ------     --------
     DSP01                                     19             0            1            0         2          1.3
     DSP02                                     23             6            3            1         2          2.5
     DSP03                                     22             1            1            2         8           .9
     DSP04                                    162             9            5            6        16           .8
     DSP05                                     84            13            8            0         5           .7
                                                                           .
                                                                           .
                                                                           .
                                                                           .
Total Responses                               365            30           22            9        34          1.0
```

Figure 59. Local Work Stations – Response Time Buckets

Remote Work Stations – Response Time Buckets – Sample:

```
                                            Component Report                                          4/05/95 15:00:39
                                  Remote Work Stations - Response Time Buckets                          Page   26
                                                 5494 RT
Member  . . . : RWSRRSPTIM  Model/Serial  . : D60     /10-15181 Main storage . . :   80.0 MB  Started . . . . :  04/05/95 14:27:51
  Library . . : COOK        System name . . :          RCHASM01 Version/Release  :    3/ 1.0  Stopped . . . . :  04/05/95 14:37:50

   Ctl/Device               Bus    IOP
---------------             ---    ---
RWS5494C01                   0      5

                                           0- 1.0      1.0- 2.0     2.0- 4.0     4.0- 8.0     > 8.0     Rsp Time
                                           ------     ---------    ---------    ---------    ------     --------
     RWS5494D00                                 2             6            4            5         2          3.8


Total Responses                                 2             6            4            5         2          3.8


Ctl                         --  Controller identifier
Device                      --  Device identifier
Bus                         --  Bus identification number
IOP                         --  Input/Output processor identification number
0- 1.0                      --  Number of response times in this range
 1.0- 2.0                   --  Number of response times in this range
 2.0- 4.0                   --  Number of response times in this range
 4.0- 8.0                   --  Number of response times in this range
> 8.0                       --  Number of response times in this range
Rsp time                    --  Average external response time (in seconds)
                                for this workstation(s)
```

V3R1 example

Figure 60. Remote Work Stations – Response Time Buckets

Exception Occurrence Summary and Interval Counts – Sample:

```
                                              Component Report                                      04/20/90 10:06:31
                                 Exception Occurrence Summary and Interval Counts                          Page 0013
                                             Sample Component Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :    96.0 M  Started . . . . :   3/29/90 11:35:54
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :    2/ 2.0  Stopped . . . . :   3/29/90 13:35:39
```

Exception Counts

Exception Type	Description	Total
Size	Size	3
Binary Overflow	Binary overflow	0
Decimal Overflow	Decimal overflow	0
Flp Overflow	Floating point overflow	0
Decimal Data	Decimal data	70
Aut Lookup	Authority lookup	7,649
PAG Fault	Process Access Group fault	0
Seize Conflict	Seize conflict	31
Lock Conflict	Lock conflict	50
Verify	Verify	549
EAO Total	Effective Address Overflow total	1,810

Exceptions per Second

Itv End	Size	Binary Overflow	Decimal Overflow	Flp Overflow	Decimal Data	Aut Lookup	PAG Fault	Seize Conflict	Lock Conflict	Verify	EAO Total
11:40	.0	.0	.0	.0	.0	3.1	.0	.8	.1	1.2	3.0
11:45	.0	.0	.0	.0	.0	5.9	.0	.1	.0	.2	.3
11:50	.0	.0	.0	.0	.0	.7	.0	.6	.2	.9	.9
11:55	.0	.0	.0	.0	.0	.1	.0	.0	.1	.1	.4

.
.
.
.

Figure 61. Exception Occurrence Summary and Interval Counts

Data Base Journaling Summary – Sample:

```
                                                 Component Report                                              3/22/95  9:49:32
                                           Data Base Journaling Summary                                               Page   22
                                        Remote SQL,ODBC: 2 WIN 3.1 Clients
Member  . . . : RODBCSQL    Model/Serial  . : F25     /10-43592  Main storage . . :   48.0 MB  Started . . . . :  03/21/95 13:47:59
  Library . . : COOK        System name . . :          RCHASM03  Version/Release  :    3/ 1.0  Stopped . . . . :  03/21/95 14:26:47
       ------- Journal Operations ------   --- Journal Deposits ---   Bundle    Bundle    --Exposed AP --   --Est Exposr --
 Itv   User      User      System    System    User      System    System    Writes    Writes    System    Not       Curr      AP Not    SMAPP
 End   Starts    Stops     Starts    Stops     Total     Total     ToUser    User      System    Jrnld     Jrnld     System    Jrnld     ReTune
-----  ------    ------    ------    ------    ------    ------    ------    ------    ------    ------    ------    ------    ------    ------
13:52      16        16         0         0      410         0         0      194         0         0        23         1         1         0
13:57      12        12         0         0      350         0         0      171         0         0        21         1         1         0
14:03      16        16         0         0      436         0         0      209         0         0        21         1         1         0
14:07      12        12         0         0      327         0         0      158         0         0        21         1         1         0
14:12      16        16         0         0      436         0         0      212         0         0        20         1         1         0
14:17      15        15         0         0      394         0         0      188         0         0        20         1         1         0
14:22      13        13         0         0      369         0         0      180         0         0        22         1         1         0
14:26      12        12         0         0      327         0         0      156         0         0        20         1         1         0
Itv End                 --  Interval end time (hour and minute)
User Starts             --  Start journal operations initiated by user
User Stops              --  Stop journal operations initiated by user
System Starts           --  Start journal operations initiated by system
System Stops            --  Stop journal operations initiated by system
User Total              --  Journal deposits resulting from user journaled
                              objects
System Total            --  Journal deposits resulting from system journaled
                              objects (total)
System ToUser           --  Journal deposits resulting from system journaled
                              objects to user created journals
Bundle Writes User      --  Bundle writes to user created journals
Bundle Writes System    --  Bundle writes to internal system journals
Exposed AP System Jrnld --  Exposed access paths currently being journaled
                              by the system
Exposed AP Not Jrnld    --  Exposed access paths currently not being journaled
Est Exposr Curr System  --  System estimated access path recovery time
                              exposure in minutes
Est Exposr AP Not Jrnld --  System estimated access path recovery time
                              exposure in minutes if no access
                              paths were being journaled by the system
SMAPP ReTune            --  System Managed Access Path Protection tuning
                              adjustments
V3R1 example
```

Figure 62. Data Base Journaling Summary

Report Selection Criteria – Sample:

```
                                              Component Report                                                   04/20/90 10:06:31
                                          Report Selection Criteria                                                     Page 0014
                                           Sample Component Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :    96.0 M  Started . . . . :   3/29/90 11:35:54
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :    2/ 2.0  Stopped . . . . :   3/29/90 13:35:39

Select Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting           Payroll              Research
                              Development          ProjectX             MrNolansStaff

                            - No Select parameters were chosen.

Omit Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting           Payroll              Research
                              Development          ProjectX             MrNolansStaff

                            - No Omit parameters were chosen.
```

Figure 63. Report Selection Criteria

B.29 Sample Transaction Reports

Job Summary – Sample:

Job Summary Report 05/17/90 4:06:07
Job Summary Page 0001

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

Job Name	User Name	*On/Off* Job Number	Pl	Typ	Pty	Prg	Tot Nbr Tns	Response Sec Avg	Response Sec Max	CPU Sec Util	CPU Sec Avg	CPU Sec Max	Average DIO/Transaction Synchronous DBR	Synchronous NDBR	Synchronous Wrt	Synchronous Sum	Synchronous Max	Async Sum	Async Max	Number Cft Lck	Number Cft Sze	K/T /Tns Sec
SCPF	QSYS	000000	02	X	52							.52					4					
QSYSARB	QSYS	020202	02	S	00							.65					3					
QLUS	QSYS	020203	02	S	20																	
QBASE	QSYS	020205	02	M	00							1.95					42			4		
QSPL	QSYS	020207	02	M	00																	
NET38	QSYS	020363	02	M	00																	
PUTGET	QSECOFR	020365	02	B	15																	
ROUTER	QSECOFR	020366	02	B	15																	
MONITOR	QSECOFR	020367	02	B	15																	
RESTART	QSECOFR	020368	02	B	16																	
SCHWEYER	SCHWEYER	020380	02	I	20																	
INSTALLE	INSTALL	020385	02	I	20																	
INSTALLE	INSTALL	020386	02	I	20																	
DRLIPPS	DRLIPPS	020391	02	I	20	Y	36	.2	3.0	.1	.08	.89		2		2	90					14
DSP01	SFRASER	020398	02	I	20	Y	204	.7	7.0	4.4	.50	6.05		2		2	42			4		11
DLH	DJW	020403	02	I	20	Y	26	.2	.9	.1	.12	.38		2		2	20					9
PRT01	QSPLJOB	020405	03	W	15																	
SFRASER	AMUNDSON	020408	02	I	20	Y	29	.5	2.7	.2	.24	1.03		7		7	103					57
MOREY	MOREY	020410	02	I	20																	
MOREY	MOREY	020412	02	I	20																	
QSNADS	QSYS	020417	02	M	00																	
QZDSTART	QSNADS	020418	02	A	40																	
QROUTER	QSNADS	020419	02	B	40																	
RCHAS374	QSNADS	020420	02	B	40																	
RCHAS374	QGATE	020421	02	B	40																	
RCHAS500	QSNADS	020422	02	B	40																	
RCHVMV	QSNADS	020423	02	B	40																	
RCHVMX2	QSNADS	020424	02	B	40																	
QDIA	QSNADS	020426	02	B	40																	
QDIALOCAL	QSNADS	020427	02	B	19																	
QNFTP	QSNADS	020429	02	B	40																	
QPFRMON	QPGMR	020437	02	B	00					.2		4.78					90					
DSP02	DRLIPPS	*020438*	02	I	20	Y	42	.4	3.2	.2	.14	.55		4	2028	2032	5167	56	304			2
PRTTNSRPT	SFRASER	020439	02	B	50					.6		14.83					311		4			
PRTTNSRPT	SFRASER	020440	02	B	50					.5		12.59					183		304			
QDFTJOBD	SFRASER	020441	02	B	50					29.0		677.63					8		304			
PRTTNSRPT	SFRASER	020442	02	B	50					.5		13.00					195		4			

Figure 64. Job Summary: Job Summary

System Summary Data (First Part) – Sample:

```
                                                    Job Summary Report                                            05/17/90  4:06:07
                                                    System Summary Data                                                   Page 0002

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

TRACE PERIODS FOR TRACE DATE.
                                  Elapsed
      Started         Stopped     Seconds
      --------        --------   ---------
      14.48.28        15.27.21       2,332

CPU BY PRIORITY FOR ALL JOBS FOR TOTAL TRACE PERIOD.

                      CPU       Cum CPU      CPU
 Pty        CPU       Util        Util        QM
----     ---------   -----      -------     ------
 00         25.353    1.08        1.08        1.01
 15                               1.08        1.01
 16                               1.08        1.01
 19                               1.08        1.01
 20        121.567    5.21        6.29        1.06
 40                               7.58        1.08
 50        718.033   30.79       38.37        1.62
 52           .519     .02       67.95        3.12
```

Figure 65. Job Summary: System Summary Data - 1

System Summary Data (Second Part) – Sample:

```
                                                    Job Summary Report                                            05/17/90  4:06:07
                                                   System Summary Data                                                    Page 0003

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

CPU AND DISK I/O PER JOB TYPE FOR ALL JOBS FOR TOTAL TRACE PERIOD.

      Job          Nbr       CPU        CPU     --Disk I/O Requests--     CPU Sec/        Sync I/O
      Type         Jobs     Seconds     Util      Sync        Async       Sync DIO        /Elp Sec        Planning Parameters
---------------    ----     -------     ----    --------     --------     --------        --------        ------------------------
INTERACTIVE          10       121.6      5.2        6156          543        .0000          6855.1        Elapsed Seconds =   2332
BATCH A,B,C,D,X      20       723.3     31.0        2381           25        .0000          7419.5        Tns Selected    =    337
SPOOL WTR/RDR         1          .0       .0           0          999        .0000              .0        TCPU=  .3607   TDIO= 315
SYSTEM JOBS           6         2.6       .1          45           84        .0578              .0        SCPU=  .0610   SDIO=5762
SYSTEM TASKS         84        17.9       .8        1634         1200        .0000          1128.5        BCPU=  .0000   BDIO= 420
---------------    ----     -------     ----    --------     --------     --------        --------        XSUM=          PDIO=   0
   ** TOTALS **     121       865.4     37.1       10216         2851        .0000          6586.7        100.00 Percent Selected

DATA FOR SELECTED TIME INTERVAL (OR TOTAL TRACE PERIOD IF NO TIME SELECTION).

  INTERACTIVE TRANSACTION AVERAGES BY JOB TYPE.

T                                          Avg     CPU/    ----- Sync Disk I/O Rqs/Tns -----  Async   W-I     Excp    Key/    Active  Est
y       Nbr    Nbr    Pct    Tns    Rsp     Tns     DB     DB     NDB    NDB           DIO    Wait    Wait    Think   K/T     Of
p  Prg  Jobs   Tns    Tns   /Hour  (Sec)   (Sec)   Read  Write   Read  Write    Sum   /Tns    /Tns    /Tns     /Tns   /Tns    AWS
-- ---  ----  ------ -----  -----  ------  ------  -----  -----  -----  -----  -----  -----  ------  -------  ------  ------  ---
IM YES    10     337 100.0    520    .549    .361      0    156      3    156    315    624    .000     .091  13.912  12.044    2

  EXCEPTIONAL WAIT BREAKDOWN BY JOB TYPE.

                     A-I         Short       Short        Seize       Lock        Event       Excs       EM3270      DDM Svr      Other
                     Wait        Wait        WaitX        Wait        Wait        Wait        ACTM        Wait        Wait        Wait
Type       Purge     /Tns        /Tns        /Tns         /Tns        /Tns        /Tns        /Tns        /Tns        /Tns        /Tns
----       -----    ------      ------      ------       ------      ------      ------      ------      ------      -------     ------
 IM         YES       .000        .013        .000         .000        .000        .000        .073        .000        .000        .005
```

Figure 66. Job Summary: System Summary Data - 2

System Summary Data (Third Part) – Sample:

```
                                             Job Summary Report                                              05/17/90  4:06:07
                                            System Summary Data                                                      Page 0004

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21
```

ANALYSIS BY INTERACTIVE TRANSACTION CATEGORIES.

Category	Avg CPU /Tns	CPU Util	Cum CPU Util	DB Read	DB Write	NDB Read	NDB Write	Sum	Async DIO /Tns	Nbr Tns	Pct Tns	Avg Rsp /Tns	Excp Wait /Tns	Avg K/T /Tns	Est Of AWS
				----- Sync Disk I/O Rqs/Tns -----											
VERY SIMPLE VS	.068	.5								182	54.0	.107	.034	11.344	1
** SIMPLE S	.071	.6	.6			1		1		208	61.7	.124	.037	12.140	1
-Boundary-	.161														
** MEDIUM M	.183	.2	.8			2		2		29	8.6	.302	.067	7.539	
-Boundary-	.219														
** COMPLEX X	1.014	4.3	5.1	1	526	7	526	1060	2104	100	29.7	1.506	.208	13.150	1
VERY COMPLEX VX	5.328	2.1		1		4		5		9	2.7	6.116	.519	5.227	
Total/Avg of **	.361				156	3	156	315	624	337	100.0	.549	.090	12.044	2

ANALYSIS BY INTERACTIVE RESPONSE TIME.

Category	Avg Rsp /Tns	Nbr Tns	Pct Tns	Cum Pct Tns	Avg CPU /Tns	CPU Util	Cum CPU Util	DB Read	DB Write	NDB Read	NDB Write	Sum	Async DIO /Tns	Excp Wait /Tns	Avg K/T /Tns
								----- Sync Disk I/O Rqs/Tns -----							
Sub-Second	.211	286	84.9	84.9	.132	1.6	1.6			1		1		.039	11.655
1 - 1.999 Sec	1.351	33	9.8	94.7	.852	1.2	2.8	1	1593	4	1593	3191	6372	.270	19.140
2 - 2.999 Sec	2.469	6	1.8	96.5	.710	.2	3.0	9		39		48		.396	5.004
3 - 4.999 Sec	3.501	3	.9	97.4	1.126	.1	3.1			35		35		1.085	5.542
5 - 9.999 Sec	6.116	9	2.7	100.1	5.328	2.1	5.2	1		4		5		.519	5.227
GE 10 Seconds				100.1			5.2								

ANALYSIS BY INTERACTIVE KEY/THINK TIME.

Category	Avg K/T /Tns	Nbr Tns	Pct Tns	Cum Pct Tns	Avg CPU /Tns	CPU Util	Cum CPU Util	DB Read	DB Write	NDB Read	NDB Write	Sum	Async DIO /Tns	Avg Rsp /Tns	Excp Wait /Tns
								----- Sync Disk I/O Rqs/Tns -----							
LT 2 Seconds	.961	143	42.4	42.4	.264	1.6	1.6		368	2	368	738	1472	.396	.072
2 - 14.999 Sec	5.754	151	44.8	87.2	.469	3.0	4.6	1		4		5		.732	.114
15 - 29.999 Sec	19.772	21	6.2	93.4	.346	.3	4.9			1		1		.470	.069
30 - 59.999 Sec	38.080	7	2.1	95.5	.342	.1	5.0			1		1		.463	.085
60 - 299.999 Sec	122.555	12	3.6	99.1	.239	.1	5.1			1		1		.353	.059
GE 300 Seconds	509.881	3	.9	100.0	.142		5.1			2		2		.246	.044

Figure 67. Job Summary: System Summary Data - 3

Distribution of Simple, Medium, and Complex Processing Unit Transactions – Sample:

```
                                                       Job Summary Report                                                  05/17/90  4:06:07
                                         Distribution of Transactions by CPU/Transaction                                           Page 0005

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

Number
 Of
 Tns
------
   50+    S
   48+    S
   46+    S
   44+    S
   42+    S
   40+    S
   38+    S
   36+    S
   34+    S S
   32+    S S
   30+    S S
   28+    S S
   26+    S S
   24+    S S
   22+    SSS S
   20+    SSS S
   18+    SSSSS
   16+    SSSSS   S                                                             X
   14+    SSSSS   S                                                             X
   12+   SSSSSS   S                                                             X
   10+   SSSSSSS  Sm                                                            X
    8+   SSSSSSSSSSm  m                                                         X
    6+   SSSSSSSSSSmm mX                                                        X
    4+   SSSSSSSSSSmm mXX   X    X X                       X                   X
    2+   SSSSSSSSSSmm mXX XXX XXXXXXXXX   X  XX            XX X X X            X
    2-   SSSSSSSSSSmmmmXXXXXXXXXXXXXXXXX XX XXXX        X   XX X X X    XX XX   X
        +----+----+----+----+----+----+----+----+----+----+----+----+----+----+ +++
     .000      .150      .300      .450      .600      .750      .900     1.050
                                CPU/Transaction  (Seconds)

Transaction Categories:

     S = Simple Transactions
     m = Medium Transactions
     X = Complex Transactions
```

Figure 68. Job Summary: Distribution of Processing Unit Transactions

Transaction Significance – Sample:

```
                                                  Job Summary Report                                        05/17/90  4:06:07
                                               Transaction Significance                                             Page 0006

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

Percent
  CPU
-------
  2.6+                                                                               X
  2.5+                                                                               X
  2.4+                                                                               X
  2.3+                                                                               X
  2.1+                                                                               X
  2.0+                                                                               X
  1.9+                                                                               X
  1.8+                                                                               X
  1.7+                                                                               X
  1.6+                                                                               X
  1.5+                                                                               X
  1.4+                                                                               X
  1.3+                                                                               X
  1.2+                                                                               X
  1.1+                                                                               X
  1.0+                                                                               X
   .9+                                                                               X
   .8+                                                                               X
   .7+                                                                               X
   .6+                                                                               X
   .5+                                                                               X
   .4+                                                                               X
   .3+                                                                               X
   .2+                                                                               X
   .1+           S                                                   X  X            X
   .1-   SSSSSSSSSSmmmmXXXXXXXXXXXXXXXXX XX XXXX        X   XX X X X    XX XX    X
        +----+----+----+----+----+----+----+----+----+----+----+----+----+----+ +++
     .000      .150      .300      .450      .600      .750      .900     1.050
                                CPU/Transaction  (Seconds)

Transaction Categories:

     S = Simple Transactions
     m = Medium Transactions
     X = Complex Transactions
```

Figure 69. Job Summary: Transaction Significance

Interactive Transactions by 5 Minute Intervals – Sample:

```
                                                          Job Summary Report                                             05/17/90  4:06:07
                                              Interactive Transactions by 5 Minute Intervals                                  Page 0007

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

                             --- Pct Of Tns ---  Pct CPU By  Nbr   Nbr    Sync  Async     Avg      Excp    Pct    Seize  Active  Est
 Itv   Active   Nbr    Tns   --- Categories ---  Categories  Sign  Sign   DIO    DIO      Rsp      Wait    Ex-Wt  Wait    K/T    Of
 End    Jobs    Tns   /Hour  %VS* %S %M %X *%VX   %S %M %X   offs  ons    /Tns   /Tns     /Tns     /Tns   /Rsp   /Tns    /Tns   AWS
-----  ------  -----  -----  ---- --+--+-- ----  ---+--+---  ----  ----  -----  -----  -------  -------  -----  ------  ------  ---
 ***
14.45*      1      9    108   44* 56+00+44  00    00+00+01                  35     28     .550     .124     23           4.333
14.50       4     65    780   52* 66+06+28  05    01+00+08                 999   1999     .664     .110     17          16.569    3
14.55       2     56    672   80* 84+02+14  05    01+00+08                  28    659     .522     .037      7           5.839    1
15.00       2     48    576   52* 54+13+33 *02    01+00+05                 768   7689     .500     .032      6          16.104    2
15.05       3     99   1188   49* 60+11+29 *02    01+01+08      1      1 78999    886     .554     .116     21           3.960    1
15.10       1     15    180   53* 53+07+40 *00    00+00+01                  19     58     .406     .124     31          19.000
15.15       2     17    204   29* 35+06+59 *00    00+00+01                 768   9877     .676     .174     26          32.882    1
15.20       1      8     96   63* 75+12+13 *00    00+00+00                6577    453     .268     .057     21          39.875    1
15.25*      1     20    240   35* 40+20+40 *00    00+00+01                  45     67     .451     .083     18           6.050
     * Denotes Partial Interval Data
```

Figure 70. Job Summary: Interactive Transactions by 5 Minute Intervals

Interactive Throughput by 5 Minute Intervals – Sample:

```
                                                          Job Summary Report                                             05/17/90  4:06:07
                                               Interactive Throughput by 5 Minute Intervals                                   Page 0008

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

                            Number Of Transactions Per Hour
 Itv
 End
        0       1000      2000      3000      4000      5000      6000      7000
-----   +----+----+----+----+----+----+----+----+----+----+----+----+----+----+
 ***
14.50    XXmSSSSS
14.55    XSSSSSS
15.00    XXmSSS
15.05    XXXmmSSSSSSS
15.10    XS
15.15    XS
15.20    S

Throughput Components:

     S = Simple Transactions
     m = Medium Transactions
     X = Complex Transactions
```

Figure 71. Job Summary: Interactive Throughput by 5 Minute Intervals

Interactive CPU Utilization by 5 Minute Intervals – Sample:

```
                                                              Job Summary Report                                             05/17/90  4:06:07
                                              Interactive CPU Utilization by 5 Minute Intervals                                      Page 0009

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :    96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :    2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

                      Percent CPU Utilization
 Itv
 End    0   10   20   30   40   50   60   70   80   90  100
-----   +----+----+----+----+----+----+----+----+----+----+
 ***
14.50    XXXXS
14.55    XXXX
15.00    XXS
15.05    XXXXS
15.10    S
15.15    X
15.20    S

CPU Components:

     S = Simple Transactions
     m = Medium Transactions
     X = Complex Transactions
```

Figure 72. Job Summary: Interactive CPU Utilization by 5 Minute Intervals

Interactive Response Time by 5 Minute Intervals – Sample:

```
                                                      Job Summary Report                                        05/17/90  4:06:07
                                       Interactive Response Time by 5 Minute Intervals                                   Page 0010

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21
                           Average Response Time (Seconds)
 Itv
 End    0      1.00      2.00      3.00      4.00      5.00      6.00      7.00
-----   +----+----+----+----+----+----+----+----+----+----+----+----+----+----+
 ***
14.50    RRRRRRw
14.55    RRRRR
15.00    RRRRR
15.05    RRRRRw
15.10    RRRw
15.15    RRRRRww
15.20    RRw

Response Components:

     R = CPU + Disk + Wait-to-Ineligible
     w = Exceptional Wait
```

Figure 73. Job Summary: Interactive Response Time by 5 Minute Intervals

Scatter Diagram of Interactive Transactions by 5 Minute Intervals – Sample:

```
                                                      Job Summary Report                                        05/17/90  4:06:07
                                Scatter Diagram of Interactive Transactions by 5 Minute Intervals                        Page 0011

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21

                    Response Time vs Number of Transactions per Hour
                    ------------------------------------------------
    OVFLW
    -----
    7.00+
        -
 R      -
 E      -
 S  6.00+
 P      -
 O      -
 N      -
 S  5.00+
 E      -
        -
 T      -
 I  4.00+
 M      -
 E      -
        -
 I  3.00+
 N      -
        -
 S      -
 E  2.00+
 C      -
 O      -
 N      -
 D  1.00+
 S      - 1     1
        - 1  011    1
        -1
    0.00+----+----+----+----+----+----+----+----+----+----+----+----+----+----+|O
        0       1000      2000      3000      4000      5000      6000      7000|F

                            Number of Transactions per Hour

 Legend:     1-9 Indicates the Number of Occurrences.
              *  Indicates more than 9 Occurrences.
              O  Identifies Average of All Occurrences.
```

Figure 74. Job Summary: Interactive Transactions by 5 Minute Intervals

Interactive Program Statistics – Sample:

Job Summary Report 05/17/90 4:06:07
Interactive Program Statistics Page 0017

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

Rank	Number Tns	Program Name	CPU /Tns	CPU Util	Cum CPU Util	Sync Disk I/O Rqs/Tns DB Read	DB Write	NDB Read	NDB Write	Sum	Async DIO /Tns	Rsp /Tns	Short Wait /Tns	Seize Wait /Tns	Pct Tns	Cum Pct Tns
1	96	QUIINMGR	.300	1.2	1.2	1		5		6		.570	.016		28.5	28.5
2	87	QSUBLDS	.757	2.8	4.1			1		1		.871	.001		25.8	54.3
3	67	QSMCSMSU	.062	.2	4.2			1		1		.163	.010		19.9	74.2
4	35	QMHGSD	.352	.5	4.8			3		3		.683	.025		10.4	84.6
5	22	QSPDSPF	.175	.2	4.9	1		4		5		.321			6.5	91.1
6	17	QPTPRCSS	.226	.2	5.1			3		3		.355	.020		5.0	96.1
7	5	QMHDSMSS	.279	.1	5.2			4		4		.718	.092		1.5	97.6
8	3	QMHDSPJL	.092		5.2							.188			.9	98.5
9	2	QSUEXIT	.066		5.2			4		4		.181			.6	99.1
10	1	*SIGNOFF*	.468		5.2			3		3		.831	.024		.3	99.4
11	1	QMHDSEXT	.352		5.2		76	15	76	167	152	1.052	.111		.3	99.7
12	1	QSMTAPS	.076		5.2			4		4		.309	.089		.3	100.0

Figure 75. Job Summary: Interactive Program Statistics

Summary of Seize/Lock Conflicts by Object – Sample:

Job Summary Report 05/17/90 4:06:07
Summary of Seize/Lock Conflicts by Object Page 0029

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

				Interactive Waiters				Noninteractive Waiters			
				Locks		Seizes		Locks		Seizes	
Type	Library	File	Member	Number	Avg Sec	Number	Avg Sec	Number	Avg Sec	Number	Avg Sec
JOBQ	QGPL	QBATCH						4	.004		
MSGQ	QUSRSYS	SFRASER		4	.005						
* Total Conflicts and Avg Sec/Conflict				4	.005			4	.004		
* Total Transactions With Conflicts				4							
* Averages Per Conflict Transaction				1.00	.005						

Figure 76. Job Summary: Summary of Seize/Lock Conflicts by Object

Priority-Jobtype-Pool Statistics – Sample:

```
                                                       Job Summary Report                                              05/17/90  4:06:07
                                                 Priority-Jobtype-Pool Statistics                                             Page 0012

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21
```

Pty	Job Type	Pool	CPU Seconds	--- Disk I/O Requests ---- Sync	Async	Number Tns
00	B	02	4.782	90		
00	M	02	1.952	42		
00	S	02	.648	3		
00	L	01	16.006	612		
00	L	02	1.965	31022	61824	
15	B	02				
15	W	03				
16	B	02				
19	B	02				
20	I	02	121.567	2356	304	336
20	S	02				
40	A	02				
40	B	02				
50	B	02	718.033	22287	41216	
52	X	02	.519	4		

Figure 77. Job Summary: Priority-Jobtype-Pool Statistics

Job Statistics – Sample

```
                                                       Job Summary Report                                              05/17/90  4:06:07
                                                         Job Statistics                                                       Page 0013

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21
```

JOBS WITH MOST TRANSACTIONS

Rank	Job Name	User Name	Job Number	Pl	Typ	Pty	Nbr Tns	Rsp /Tns	CPU /Tns	CPU Util	Cum CPU Util	Sync DIO /Tns	Async DIO /Tns	Nbr W-I	Nbr A-I	Nbr Evt	Number Conflict Lck	Sze	Pct Tns	Cum Pct Tns
1	DSP01	SFRASER	020398	02	I	20	204	.699	.504	4.4	4.4	2					4		60.5	60.5
2	DSP02	DRLIPPS	020438	02	I	20	42	.379	.139	.3	4.7	127	245						12.5	73.0
3	DRLIPPS	DRLIPPS	020391	02	I	20	36	.195	.080	.1	4.8	3							10.7	83.7
4	SFRASER	AMUNDSON	020408	02	I	20	29	.461	.239	.3	5.1	8							8.6	92.3
5	DLH	DJW	020403	02	I	20	26	.239	.122	.1	5.2	3							7.7	100.0
6																				
7																				

Figure 78. Job Summary: Job Statistics

Interactive Program Statistics – Sample

```
                                                       Job Summary Report                                              05/17/90  4:06:07
                                                 Interactive Program Statistics                                               Page 0018

Member  . . . : TJST41       Model/Serial  . . :  B60/10-12883   Main storage . . :   96.0 M   Started  . . . . : 05/16/90 14:48:28
  Library . . : TJSWORK2     System name . . . :      RCH38366   Version/Release  :   2/ 2.0   Stopped  . . . . : 05/16/90 15:27:21
```

PROGRAMS WITH HIGHEST CPU/TNS

Rank	Number Tns	Program Name	CPU /Tns	CPU Util	Cum CPU Util	----- Sync Disk I/O Rqs/Tns ----- DB Read	DB Write	NDB Read	NDB Write	Sum	Async DIO /Tns	Rsp /Tns	Short Wait /Tns	Seize Wait /Tns	Pct Tns	Cum Pct Tns
1	87	QSUBLDS	.757	2.8	2.8			1		1		.871			25.8	25.8
2	1	*SIGNOFF*	.468		2.8			3		3		.831			.3	26.1
3	35	QMHGSD	.352	.5	3.4			3		3		.683			10.4	36.5
4	1	QMHDSEXT	.352		3.4		76	15	76	167	152	1.052			.3	36.8
5	96	QUIINMGR	.300	1.2	4.6	1		5		6		.570			28.5	65.3
6	5	QMHDSMSS	.279	.1	4.7			4		4		.718			1.5	66.8
7	17	QPTPRCSS	.226	.2	4.8			3		3		.355			5.0	71.8
8	22	QSPDSPF	.175	.2	5.0	1		4		5		.321			6.5	78.3
9	3	QMHDSPJL	.092		5.0							.188			.9	79.2
10	1	QSMTAPS	.076		5.0			4		4		.309			.3	79.5

Figure 79. Job Summary: Interactive Program Statistics

Individual Transaction Statistics – Sample

Job Summary Report 05/17/90 4:06:07
Individual Transaction Statistics Page 0021

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2 /2.0 Stopped . . . . : 05/16/90 15:27:21

TRANSACTIONS WITH LONGEST RESPONSE TIMES

Rank	Value	Time	Program	Job Name	User Name	Number	Pool	Type	Priority
1	6.953	14.51.51.619	QSUBLDS	DSP01	SFRASER	020398	02	I	20
2	6.859	15.05.57.610	QMHGSD	DSP01	SFRASER	020398	02	I	20
3	6.684	14.54.45.748	QSUBLDS	DSP01	SFRASER	020398	02	I	20
4	6.485	14.57.04.456	QSUBLDS	DSP01	SFRASER	020398	02	I	20
5	6.008	14.58.39.300	QSUBLDS	DSP01	SFRASER	020398	02	I	20
6	5.691	15.05.08.794	QSUBLDS	DSP01	SFRASER	020398	02	I	20
7	5.620	15.02.55.162	QSUBLDS	DSP01	SFRASER	020398	02	I	20
8	5.375	14.53.17.618	QSUBLDS	DSP01	SFRASER	020398	02	I	20
9	5.367	14.55.59.637	QSUBLDS	DSP01	SFRASER	020398	02	I	20
10	3.830	14.51.24.024	QSUBLDS	DSP01	SFRASER	020398	02	I	20

Figure 80. Job Summary: Individual Transaction Statistics

Longest Seize/Lock Conflicts – Sample:

Job Summary Report 05/17/90 4:06:07
Longest Seize/Lock Conflicts Page 0027

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

Rank	Value	Time	Job Name	User Name	Job Number	Pl	Typ	Pty	S/L	Holder- / Object-	Job Name.. / Type..	User Name. / Library...	Number / File......	Pool / Member....	Type / RRN......	Pty
1	.006	15.10.13.223	DSP01	SFRASER	020398	02	I	20	L	HOLDER-	PRTTNSRPT	SFRASER	020439	02	I	20
										OBJECT-	MSGQ	QUSRSYS	SFRASER			
2	.005	15.25.20.550	DSP01	SFRASER	020398	02	I	20	L	HOLDER-	PRTTNSRPT	SFRASER	020442	02	I	20
										OBJECT-	MSGQ	QUSRSYS	SFRASER			
3	.005	15.23.22.018	DSP01	SFRASER	020398	02	I	20	L	HOLDER-	QDFTJOBD	SFRASER	020441	02	I	20
										OBJECT-	MSGQ	QUSRSYS	SFRASER			
4	.005	15.10.21.028	DSP01	SFRASER	020398	02	I	20	L	HOLDER-	PRTTNSRPT	SFRASER	020440	02	I	20
										OBJECT-	MSGQ	QUSRSYS	SFRASER			
5	.005	15.10.01.483	QBASE	QSYS	020205	02	M	00	L	HOLDER-	DSP01	SFRASER	020398	02	I	20
										OBJECT-	JOBQ	QGPL	QBATCH			
6	.005	15.08.06.635	QBASE	QSYS	020205	02	M	00	L	HOLDER-	DSP01	SFRASER	020398	02	I	20
										OBJECT-	JOBQ	QGPL	QBATCH			
7	.004	15.25.00.872	QBASE	QSYS	020205	02	M	00	L	HOLDER-	DSP01	SFRASER	020398	02	I	20
										OBJECT-	JOBQ	QGPL	QBATCH			
8	.004	15.11.17.183	QBASE	QSYS	020205	02	M	00	L	HOLDER-	DSP01	SFRASER	020398	02	I	20
										OBJECT-	JOBQ	QGPL	QBATCH			

Figure 81. Job Summary: Longest Seize/Lock Conflicts

Longest Holders of Seize/Lock Conflicts – Sample:

Job Summary Report 05/17/90 4:06:07
Longest Holders of Seize/Lock Conflicts Page 0028

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

			Job	User	Job					Object				
Rank	Value	Time	Name	Name	Number	Pl	Typ	Pty	S/L	Type	Library	File	Member	RRN
1	.004	15.11.17.183	QBASE	QSYS	020205	02	M	00	L	JOBQ	QGPL	QBATCH	QNETF	

Figure 82. Job Summary: Longest Holders of Seize/Lock Conflicts

Batch Job Analysis – Sample:

Job Summary Report 05/17/90 4:06:07
Batch Job Analysis Page 0030

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

Job Name	User Name	Job Number	Pl	Typ	Pty	Start	Stop	Elapsed Seconds	CPU Seconds	CPU Util	Sync Disk I/O	Async Disk I/O	BCPU /DIO	--- Synchronous --- --DIO/Sec-- Elp	Act	Ded	Excp Wait Sec
PRTTNSRPT	SFRASER	020439	02	B	50	15.08.06	15.10.14	127.725	14.826	11.6	5311	325		41.6	209	30	102.318
PRTTNSRPT	SFRASER	020440	02	B	50	15.10.01	15.10.21	19.752	12.586	63.7	183	34		134	376	30	.157
QDFTJOBD	SFRASER	020441	02	B	50	15.11.17	15.23.22	724.887	677.625	93.5	6598	10304		77	90	30	.042
PRTTNSRPT	SFRASER	020442	02	B	50	15.25.01	15.25.21	19.912	12.996	65.3	195	23		710	961	30	.132

Figure 83. Job Summary: Batch Job Analysis

Batch Thread Analysis – Sample:

Job Summary Report 05/17/90 4:06:07
Batch Thread Analysis Page 0031

Member . . . : TJST41 Model/Serial . . : B60/10-12883 Main storage . . : 96.0 M Started . . . . : 05/16/90 14:48:28
Library . . : TJSWORK2 System name . . . : RCH38366 Version/Release : 2/ 2.0 Stopped . . . . : 05/16/90 15:27:21

Thread	Pty	Number Jobs	Elapsed Seconds	CPU Seconds	Excp Wait	Sync Disk I/O	Async Disk I/O
1	00	1	2332.668	4.782	2320.389	654	435
2	15	1	2332.675		2332.675	654	765
3	15	1	2332.677		2332.677	876	876
4	15	1	2332.684		2332.684	467	654
5	16	1	2332.676		2332.676	988	877
6	19	1	2332.665		2332.665	5463	546
7	40	1	2332.664		2332.664	765	876
8	40	1	2332.665		2332.665	979	765
9	40	1	2332.668		2332.668	243	424
10	40	1	2332.668		2332.668	877	453
11	40	1	2332.669		2332.669	456	776
12	40	1	2332.669		2332.669	788	876
13	40	1	2332.673		2332.673	344	243
14	40	1	2332.675		2332.675	543	5456
15	50	1	19.752	12.586	.157	5183	10304
16	50	3	872.524	705.447	102.492	34104	430912

Figure 84. Job Summary: Batch Thread Analysis

Report Selection Criteria – Sample:

```
                                                        Job Summary Report                                        04/20/90 10:06:31
                                                     Report Selection Criteria                                          Page 0014
                                                     Sample Transaction Report

Member  . . . : GOODSTUF     Model/Serial  . . :  B60/10-12883   Main Storage . . :     96.0 M  Started . . . . :   3/29/90 11:35:54
  Library . . : QPFRDATA     System name . . . :  RCH38366       Version/Release  :     2/ 2.0  Stopped . . . . :   3/29/90 13:35:39

Select Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - Jobname1
                              Jobname2
                              Jobnum

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Functional Areas            - Accounting           Payroll              Research
                              Development          ProjectX             MrNolansStaff

                            - No Select parameters were chosen.

Omit Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Functional Areas            - Accounting           Payroll              Research
                              Development          ProjectX             MrNolansStaff

                            - No Omit parameters were chosen.

Options Selected            - SS Include Special Summary Reports
```

Figure 85. Job Summary Report: Report Selection Criteria

Transaction Report Option – Sample

```
                                                        Transaction Report                                         4/08/92 12:10:58
                                                                                                                          Page 0001
Member  . . . : MAR11        Model/Serial  . . :  D80/10-00725   Main storage . . :    64.0 M   Started  . . . . : 03/11/92 08:03:58
  Library . . : DFL3C        System name . . . :      RCHAS725   Version/Release  :    2/ 2.0   Stopped  . . . . : 03/11/92 08:46:47
Job name  . . : SCPF         User name . . . . :     QSYS        Job number . . . :    000000   TDE/Pl/Pty/Prg . :    0088/02/40/NO

        E           T   CPU    ---- Physical I/O Counts ----- ***** Transaction Response Time (Sec/Tns) ****** -BMPL-
        x           y   Sec    ----- Synchronous ------ Async  ******  - Activity Level Time -  Inel    Long   C    I Seize
        c   Program p   Per     DB   DB  NDB   NDB       Disk   ****            Short   Seize   Time    Wait   u    n  Hold   Key/
  Time  p    Name   e   Tns    Read  Wrt Read  Wrt   Sum  I/O    **    Active   Wait     Cft   A-I/W-I Lck/Oth r    l  Time  Think
-------- - ---------- - ------- ---- ---- ---- ---- ---- ----- -------- ------- ------- ------- ------- ------- --   -- ----- ------
08.07.29   QWCISCFR  1    .092    2    2    2    1    7     7     .255    .255                                   1             210.5
08.10.37   QWCISCFR  1    .037              2         2     0  188.527    .151             .057         188.376  1                .0
08.13.21 Y QWCISCFR  1    .101    2    2    4    1    9     6  163.288    .290                          162.997  2                .0
08.14.32   QWCISCFR  1    .022                        0     0   71.526    .083             .059          71.443  1                .0
08.15.06   QWCISCFR  1    .052                        0     0   34.456    .120             .070          34.336  1                .0
08.22.29   QWCISCFR  1    .083    2    2         1    5     6  442.280    .170                          442.110  2                .0
08.41.02   QWCISCFR  1    .086    2    6    1    2   11     7 1112.816    .323                         1112.491  2                .0
08.45.29   QWCISCFR  1    .079    2    2         1    5     6  267.195    .243                          266.951  1                .0
08.46.43   QWCISCFR  1    .095    2    2    2    1    7     5   74.117    .684                           73.433  1                .0
-----------------------------------------------
J O B   S U M M A R Y   D A T A   (T O T A L S)
-----------------------------------------------
        Average          .072     1
                                       2    1    1    5     4  261.607    .092    .000    .062    .000 102.266            .0  210.5
        Count                                                     9       25              3               23                   1
        Minimum          .022                                      .255                    .057            .035                210.5
        Maximum          .101                        11     7 1112.816    .684             .070        1112.491                210.5
        Total/Job        .647                        46
                                                           37
                                                              2568.261 Elapsed       .0 Percent CPU Utilization
```

Figure 86. Transaction Report

Transition Report Option – Sample:

```
Member  . . . : MAR11       Model/Serial  . . :  D80/10-00725   Main storage . . :   64.0 M   Started  . . . . :03/11/92 08:03:58
  Library . . : DFL3C       System name . . . :      RCHAS725   Version/Release  :   2/ 2.0   Stopped  . . . . :03/11/92 08:46:47
Job name  . . : QPADEV0008  User name . . . . :    DFL          Job number . . . :   015921   TDE/Pl/Pty/Prg . :    0A57/03/20/YES
Job type  . . : I       Elapsed Time -- Seconds           Sync/Async Phy I/O    -MPL-
                        -----------------------        ------------------------ C  I     Last 4 Programs in Invocation Stack
             State Wait   Long   Active   Inel    CPU    DB    DB  NDB   NDB      u  n -------------------------------------------
    Time     W A I Code   Wait    /Rsp*   Wait    Sec   Read  Wrt Read  Wrt  Tot r  l    Last       Second      Third     Fourth
------------ ----- ---- ------- ------- ------- -------- ---- ---- ---- ---- ---- -- -- ---------- ---------- ---------- ----------
 ----------  QUYLIST              2.376*           .355    2    0   98    4  104*      PAG=    6  XSum=    0 PWrt=    4
                                                           1    0    0   12   13
08.46.27.576 ->A          9.797    .002            .002              1             1
                                                                                       PAG=    1  XSum=    0 PWrt=    0
08.46.27.576       SOT1                                              1             1    QPADEV0008 QDUODSPF
08.46.27.606       SOT1            .031            .004                            1    QPADEV0008 QDUODSPF
08.46.27.925   W                   .318            .083              4
                                                                          2        1    QT3REQIO   QWSCLOSE   QDMCLOSE   QUOCPP
                                                                          6             PAG=    0  XSum=    0 PWrt=    2
08.46.27.926   A                   .001            .001                            1
08.46.27.965       EOR1            .039            .036                            1    QPADEV0008 QDUI132
08.46.27.965       EOT1                                                            1    QPADEV0008 QDUI132
08.46.27.972 W<-                   .007            .007                            1    QT3REQIO   QWSGET     QUIINMGR   QUIEXFMT
 ----------  QUIINMGR              .398*           .133    0    0    6    2    8*       PAG=    1  XSum=    0 PWrt=    2
                                                           0    0    0    6    6
08.46.34.655 ->A          6.684    .001            .001                            1
08.46.34.655       SOT1                                                            1    QPADEV0008 QDUI132
08.46.34.755       EOR1            .099            .059                            1    QPADEV0008 QDUI132
08.46.34.755       EOT1                                                            1    QPADEV0008 QDUI132
08.46.34.758 W<-                   .003            .003                            1
 ----------  QUIINMGR              .020*           .015    0    0    0    0    0*
08.46.35.261      /OFF     .503                    .001
08.46.35.261 *TRACE OFF
-----------------------------------------------
J O B   S U M M A R Y   D A T A   (T O T A L S)
-----------------------------------------------
                        CPU   ---- Physical I/O Counts -----  ***** Transaction Response Time (Sec/Tns) ******  -BMPL-
                        Sec   ----- Synchronous ------ Async   ******  - Activity Level Time -  Inel    Long    C    I Seize
                        Per    DB   DB  NDB   NDB       Disk    ****            Short   Seize   Time    Wait    u    n  Hold   Key/
                        Tns   Read  Wrt Read  Wrt  Sum  I/O      **    Active   Wait     Cft  A-I/W-I Lck/Oth  r    l  Time  Think
                      ------- ---- ---- ---- ---- ---- -----  -------- ------- ------- ------- ------- -------  --  -- ----- ------
        Average          .104    0    0   17    1   18     1      .485    .088    .051    .020    .000   3.145            .0   14.8
        Count                                                  167     254      59      2               1               6    167
        Minimum          .010                                     .012                    .015           3.145                   .1
        Maximum         3.177                       996    27   23.264  12.831   2.098    .025           3.145                549.6
        Total/Job      17.318                      2887
                                                         207  2568.265 Elapsed       .7 Percent CPU Utilization
```

Figure 87. Transition Report

B.30 Sample Job Interval Reports

Interactive Job Summary – Sample:

```
                                              Job Interval Report                                    04/19/90 13:47:50
                                            Interactive Job Summary                                           Page    1
                                           User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . :  B60/10-12883   Main storage  . . :   96.0 M   Started  . . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :      TEST#366   Version/Release . :    2/2.0   Stopped  . . . . : 01/18/90 09:34:00

 Itv    Act      Tns      Rsp/   ------------- Number of I/O --------------   Tns/    CPU    PAG    EAO    XSum    Perm   Arith
 End    Jobs    Count     Tns       Sync      Async     Logical      Cmn      Hour    Util   Fault   Excp    I/O    Write  Ovrflw
-----  -----  ----------  ------  ---------  ---------  ---------  ---------  -------  ----  ------  ------  ------  ------  ------
09:19      2          58    2.43       1075          0       5361          0      239   4.7       0       0       0       0       0
09:34      2          17    1.41        207          0        775          0       68    .5       0       0       0       0       0
```

Figure 88. Interactive Job Summary Section

Noninteractive Job Summary – Sample

```
                                              Job Interval Report                                    04/19/90 13:47:50
                                           Noninteractive Job Summary                                         Page    2
                                           User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . :  B60/10-12883   Main storage  . . :   96.0 M   Started  . . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :      TEST#366   Version/Release . :    2/2.0   Stopped  . . . . : 01/18/90 09:34:00

 Itv    Act    CPU    ---- Number of I/O Per Second ----  - CPU/ I/O -   Line    Page     PAG      EAO      XSum     Perm     Arith
 End    Jobs   Util     Sync    Async   Logical     Cmn    Sync  Async   Count   Count   Fault     Excp     I/O     Write    Ovrflw
-----  -----  ----  -------  -------  -------  -------  -----  -----  -------  -----  -------  -------  -------  -------  -------
09:19      2    .3       .2       .0       .0       .0     17      0        0      0        0        0        0        0        0
09:34      2    .0       .0       .0       .0       .0     16      0        0      0        0        0        0        0        0
```

Figure 89. Noninteractive Job Summary Section

Interactive Job Detail – Sample

```
                                              Job Interval Report                                    04/19/90 13:47:50
                                            Interactive Job Detail                                            Page    3
                                           User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . :  B60/10-12883   Main storage  . . :   96.0 M   Started  . . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :      TEST#366   Version/Release . :    2/2.0   Stopped  . . . . : 01/18/90 09:34:00

                                                                             -------- Physical I/O per Transaction  ------          SYNC
 Itv       Job          User       Job                TNS     Rsp     CPU    ---- Synchronous -----  ---- Asynchronous ----  CPU      I/O
 End       Name         Name       Number  PL  Pty    /HR     /Tns    /Tns    DBR   DBW  NDBR  NDBW   DBR   DBW  NDBR  NDBW  Util    /Sec
-----  ----------  ----------  ------  --  ---  -----  -------  -------  ----  ----  ----  ----  ----  ----  ----  ----  ----  -------
09:19  DDWEBER     MEP         006209   3   20    161     2.15     .987    .5    .0  10.6    .0    .0    .0    .0    .0   4.4      1.2
09:19  NOVEY       NOVEY       006232   3   20     78     3.00     .147    .0    .0   1.0    .0    .0    .0    .0    .0    .3       .0
09:34  DDWEBER     MEP         006209   3   20     12      .66     .408    .0    .0   1.6    .0    .0    .0    .0    .0    .1       .0
09:34  NOVEY       NOVEY       006232   3   20     56     1.57     .253   3.6    .0   9.0    .0    .0    .0    .0    .0    .3       .2
```

Figure 90. Job Interval Report: Interactive Job Detail Section

Noninteractive Job Detail – Sample

```
                                         Job Interval Report                                     04/19/90 13:47:50
                                      Noninteractive Job Detail                                          Page    4
                                     User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . :  B60/10-12883   Main storage  . . :   96.0 M   Started  . . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :      TEST#366   Version/Release . :    2/2.0   Stopped  . . . . : 01/18/90 09:34:00

 Itv      Job         User       Job                               Elapsed    CPU    --- Nbr I/O /Sec ---   - CPU / I/O -  --- Printer ---
 End      Name        Name       Number     Pool    Type  Pty       Time     Util    Sync     Async   Lgl    Sync    Async   Lines    Pages
-----  ----------  ----------  ------     ----    ----  ---      -------    ----   -----    -----  -----   -----   -----  -------   -----
09:19  QNFTP       QSNADS      006181        2      B    40        14:31      .3       0        0      0      22       0        0       0
09:19  QROUTER     QSNADS      006176        2      B    40        14:31      .0       0        0      0       9       0        0       0
09:34  QNFTP       QSNADS      006181        2      B    40        14:58      .0       0        0      0      27       0        0       0
09:34  QROUTER     QSNADS      006176        2      B    40        14:58      .0       0        0      0       9       0        0       0
```

Figure 91. Job Interval Report: Noninteractive Job Detail Section

Report Selection Criteria Section – Sample

```
                                         Job Interval Report                                     04/19/90 13:28:36
                                      Report Selection Criteria                                          Page    3
                                     User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . :  B60/10-12883   Main storage  . . :   96.0 M   Started  . . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :      TEST#366   Version/Release . :    2/2.0   Stopped  . . . . : 01/18/90 09:34:00

Select Parameters

Pools                      - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                       - 012345/Useridwxyz/Jobname123
                             987654/Useridabcd/Jobname456

User IDs                   - User1      User2      User3      User4     User5      User6
                             User7      User8      User9      User10    User11     User12

Subsystems                 - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                             Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines       - Line1      Line2      Line3      Line4     Line5      Line6
                             Line7      Line8      Line9      Line10    Line11     Line12

Control Units              - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                             Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas           - Accounting           Payroll              Research
                             Development          ProjectX             MrNolansStaff

                           - No Select parameters were chosen.

Omit Parameters

Pools                      - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                       - 012345/Useridwxyz/Jobname123
                             987654/Useridabcd/Jobname456

User IDs                   - User1      User2      User3      User4     User5      User6
                             nnnnnn     User8      User9      User10    User11     User12

Subsystems                 - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                             Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines       - Line1      Line2      Line3      Line4     Line5      Line6
                             Line7      Line8      Line9      Line10    Line11     Line12

Control Units              - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                             Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas           - Accounting           Payroll              Research
                             Development          ProjectX             MrNolansStaff

                           - No Omit parameters were chosen.
```

Figure 92. Job Interval Report: Report Selection Criteria Section

B.31 Sample Pool Interval Reports

Subsystem Activity – Sample

```
                                                   Pool Interval Report                                         04/19/90 13:28:36
                                                    Subsystem Activity                                                  Page    1
                                                User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

                                         --------- Physical I/O per Transaction ---------  ------------- Job Maximums ------------
 Itv   Subsystem      CPU                ----- Synchronous -----  ---- Asynchronous -----  CPU   Phy
 End   Name        PL  Util    Tns         DBR   DBW   NDBR  NDBW   DBR   DBW   NDBR  NDBW  Util  I/O    Tns    Rsp     A-W  W-I A-I
-----  ----------  --  ----  ----------  ----- ----- ----- -----  ----- ----- ----- -----  ---- ------ ----- ------ ------ --- ---
09:19  QBASE        2    .5          32               1.5                                     .9     41    30    .16     45   0   0
09:19  QBASE        3  19.3         472     .1        7.4                                    6.1  2,975    82   3.00    113   0   0
09:19  QSNADS       2   1.1           0                                                       .3    423     0    .00    460   0   0

09:34  QBASE        2   3.2           2    6.5      176.5                                   79.1    199     1   2.00     37   0   0
09:34  QBASE        3   4.3         310     .2        2.9                                    1.4    378   121   2.25    133   0   0
09:34  QSNADS       2    .2           0                                                       .0    158     0    .00     87   0   0
```

Figure 93. Pool Interval Report: Subsystem Activity

Pool Activity – Sample

```
                                                   Pool Interval Report                                         04/19/90 13:28:36
                                                      Pool Activity                                                     Page    2
                                                User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

                                         -------- Physical I/O per Transaction ----------  ------------- Job Maximums ------------
 Itv        Act   Size  CPU              ----- Synchronous -----  ---- Asynchronous -----  CPU   Phy
 End   PL  Lvl   (K)   Util    Tns         DBR   DBW   NDBR  NDBW   DBR   DBW   NDBR  NDBW  Util  I/O    Tns    Rsp     A-W  W-I A-I
-----  --  ---  -----  ----  ----------  ----- ----- ----- -----  ----- ----- ----- -----  ---- ------ ----- ------ ------ --- ---
09:19   2    7      7   1.4          32     .1        3.2                                     .9    423    30    .16    460   0   0
09:19   3   87     77  19.3         472     .1        7.4                                    6.1  2,975    82   3.00    113   0   0

09:34   2    7      7   3.4           2    8.0      183.5                                   79.1    199     1   2.00     87   0   0
09:34   3   87     77   4.3         310     .2        2.9                                    1.4    378   121   2.25    137   0   0
```

Figure 94. Pool Interval Report: Pool Activity

Report Selection Criteria – Sample

```
                                                 Pool Interval Report                                                     04/19/90 13:28:36
                                              Report Selection Criteria                                                         Page    3
                                              User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

Select Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              User7      User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting          Payroll              Research
                              Development         ProjectX             MrNolansStaff

                            - No Select parameters were chosen.

Omit Parameters

Pools                       - 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

Jobs                        - 012345/Useridwxyz/Jobname123
                              987654/Useridabcd/Jobname456

User IDs                    - User1      User2      User3      User4     User5      User6
                              nnnnnn     User8      User9      User10    User11     User12

Subsystems                  - Subsystem1 Subsystem2 Subsystem3 Subsystem4 Subsystem5 Subsystem6
                              Subsystem7 Subsystem8 Subsystem9 Subsystema Subsystemb Subsystemc

Communications Lines        - Line1      Line2      Line3      Line4     Line5      Line6
                              Line7      Line8      Line9      Line10    Line11     Line12

Control Units               - Ctlr1      Ctlr2      Ctlr3      Ctlr4     Ctlr5      Ctlr6
                              Ctlr7      Ctlr8      Ctlr9      Ctlr10    Ctlr11     Ctlr12

Functional Areas            - Accounting          Payroll              Research
                              Development         ProjectX             MrNolansStaff

                            - No Omit parameters were chosen.
```

Figure 95. Pool Interval Report: Report Selection Criteria

B.32 Sample Resource Interval Reports

Disk Utilization Summary – Sample

```
                                                     Resource Interval Report                            04/19/90 13:58:59
                                                     Disk Utilization Summary                                     Page    1
                                                    User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

            Average        Average      Average       Average                          High       High      High       Disk
 Itv        Physical        Reads        Writes        K per       Avg        High     Util       Srv       Srv        Space
 End        I/O /Sec       /Sec          /Sec          I/O        Util       Util      Unit       Time      Unit       Used
-----      ---------     ---------     --------      -------      ----       ----      ----       ----      ----      -------
09:19           10.3           5.0          5.3          1.4        .6        1.5      0009       .039      0033        5,182
09:34            5.9           3.5          2.3          1.5        .3        1.5      0018       .049      0033        5,186
-----      ---------     ---------     --------      -------      ----
Average:         8.0           4.3          3.7          1.5        .5
```

Figure 96. Resource Interval Report: Disk Utilization Summary

Disk Utilization Detail – Sample

```
                                                     Resource Interval Report                            04/19/90 13:58:59
                                                     Disk Utilization Detail                                      Page    2
                                                    User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

        Bus    IOP Id/     ASP   CSS    Itv    ---- Physical I/O per Second ---    K per             Queue    Avg Time per  I/O
Unit    ID     Model       ID    ID     End       Total      Reads      Writes      I/O     Util     Length     Service    Wait
----    ---    ---------   ---   ---   -----   ----------  ---------  ---------   -------   ----    -------     -------    ----
0001      0    01 (6110)   01     00   09:19         .288       .095       .192       4.6    1.0        .24        .032    .003
                                       09:34         .433       .311       .132       3.1    1.1        .01        .053    .006
                                               ----------  ---------  ---------   -------   ----    -------     -------    ----
Unit Average:                                        .366       .204       .162       3.7    1.0        .12        .042    .004

0002      0     01 (6110)  01     00   09:19         .117       .050       .066       6.8     .3        .12        .016    .005
                                       09:34         .094       .060      .034        5.1     .3        .24        .023    .004
                                               ----------  ---------  ---------   -------   ----    -------     -------    ----
Unit Average:                                        .105       .055       .050       6.0     .3        .18       .0019    .004
```

Figure 97. Resource Interval Report: Disk Utilization Detail

Communications Line Detail – SDLC Sample

```
                                                   Resource Interval Report                                      04/19/90 13:58:5
                                                   Communications Line Detail                                           Page    1
                                                   User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA     System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = SDLC  (SORT BY INTERVAL)
```

Itv End	Bus/ IOP/ Line	Line Speed	Line Util	Bytes Trnsmitd Per Sec	Total IFrames Trnsmitd	Percent I Frames Trnsmitd in Error	Bytes Recd per Sec	Total Frames Recd	Percent Frames Received in Error	Pct Poll Retry Time	- Congestion -- Local Not Ready	Remote Not Ready
	Bus 0 IOP 05 (6110)											
13:08	MPLSCHI	9.6	48	384	2,645	21	384	4,086	08	03	18	00
13:28	MPLSROCH	9.6	38	384	2,645	18	384	4,086	06	02	06	00
13:48	SUPPLIER	19.2	22	384	2,645	04	384	4,086	02	18	07	02

Figure 98. Resource Interval Report: Communications Line Detail - SDLC

Communications Line Detail – X.25 Sample

```
                                                    Resource Interval Report                                    01/10/92 10:49:30
                                                   Communications Line Detail                                           Page  10
                                                   User-Selected Report Title

 Member  . . . : MONDAY       Model/Serial  . . : B70/XX-XXXXX    Main storage  . . :  16.0 M     Started . . . : 10/19/90 13:48:27
   Library . . : QPFRDATA     System name . . . :     SYS400      Version/Release . :   2/2.0     Stopped . . . : 10/19/90 14:17:02

 PROTOCOL = X.25 (SORT BY INTERVAL)
```

Itv End	Bus/ IOP/ Line	Line Speed	Transmit/ Receive/ Average Line Util	Bytes Trnsmitd Per Sec	Total I Frames Trnsmitd	Percent I Frames Trnsmitd In Error	Bytes Recd Per Sec	Total Frames Recd	Percent Frames Recd In Err	------Reset------ -----Packets----- Trnsmitd	Recd
	BUS 1 IOP 1 (6130)										
13:53	NLX25	9.6	13/01/07	160	216	0	0	14	0	42,800	0
13:58	NLX25	9.6	11/00/05	132	156	0	0	8	0	31,100	0
14:03	NLX25	9.6	16/02/09	196	250	0	0	28	0	49,500	0

Figure 99. Resource Interval Report: Communications Line Detail - X.25

Communications Line Detail – TRLAN Sample

```
                                                   Resource Interval Report                                     04/19/90 13:58:59
                                                   Communication Line Detail                                            Page    1
                                                   User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA     System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = TRLAN (SORT BY INTERVAL)
```

Itv End	Bus/ IOP/ Line	Line Speed	Line Util	I/Frames Trnsmitd per Sec	I/Frames Recd per Sec	------- Congestion ------- -- Local --- Not Ready	Seq Error	-- Remote -- Not Ready	Seq Error	Frame Retry	Rsp Timer Ended	Remote LAN --- Frames --- Trnsmitd	Recd	MAC Errors
	Bus 1 IOP 01 (2619)													
13:08	LOCALNET	4000.0	78	382	384	10	8	22	9	4	10	18	17	112
13:23	LOCALNET	4000.0	78	382	384	10	8	22	9	4	10	18	17	110
13:38	LOCALNET	4000.0	78	382	384	10	8	22	9	4	10	18	17	99

Figure 100. Resource Interval Report: Communications Line Detail - TRLAN

Communications Line Detail – ELAN Sample

```
                                                    Resource Interval Report                                      04/19/90 13:58:59
                                                    Communication Line Detail                                            Page     1
                                                    User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA     System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = ELAN (SORT BY INTERVAL)

                                                          ------- Congestion -------
        Bus/                            I/Frames    I/Frames   -- Local ---  -- Remote --              Rsp
 Itv    IOP/      Line      Line        Trnsmitd     Recd       Not    Seq    Not    Seq    Frame    Timer
 End    Line      Speed     Util        per Sec     per Sec     Ready  Error  Ready  Error  Retry    Ended
-----   ---------  --------  ----     ---------   ---------    -----  -----  -----  -----  ------   ------
        Bus 1
        IOP 01
         (2617)

13:08   LOCALNET2    4000.0    78          382         384       10      8     22      9       4       10
13:23   LOCALNET2    4000.0    78          382         384       10      8     22      9       4       10
13:38   LOCALNET2    4000.0    78          382         384       10      8     22      9       4       10
```

Figure 101. Resource Interval Report: Communications Line Detail – ELAN

Communications Line Detail – DDI Sample

```
                                                    Resource Interval Report                                      04/19/90 13:58:59
                                                    Communication Line Detail                                            Page     1
                                                    User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA     System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = DDI (SORT BY INTERVAL)

                                                          ------- Congestion -------
        Bus/                            I/Frames    I/Frames   -- Local ---  -- Remote --              Rsp
 Itv    IOP/      Line      Line        Trnsmitd     Recd       Not    Seq    Not    Seq    Frame    Timer    MAC
 End    Line      Speed     Util        per Sec     per Sec     Ready  Error  Ready  Error  Retry    Ended    Errors
-----   ---------  --------  ----     ---------   ---------    -----  -----  -----  -----  ------   ------   ------
        Bus 1
        IOP 01
         (2618)

13:08   LOCALNET2  100000.0    78          842         333       10      8     22      6       4       11       20
13:23   LOCALNET2  100000.0    79          867         399       13      7     23      5       4       29       33
13:38   LOCALNET2  100000.0    82          997         522       23      8     32      7       7       52      123
```

Figure 102. Resource Interval Report: Communications Line Detail – DDI

Communications Line Detail – FRLY Sample

```
                                                    Resource Interval Report                                      04/19/90 13:58:59
                                                    Communication Line Detail                                            Page     1
                                                    User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA     System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = FRLY (SORT BY INTERVAL)

                                                          ------- Congestion -------
        Bus/                            I/Frames    I/Frames   -- Local ---  -- Remote --              Rsp
 Itv    IOP/      Line      Line        Trnsmitd     Recd       Not    Seq    Not    Seq    Frame    Timer    MAC
 End    Line      Speed     Util        per Sec     per Sec     Ready  Error  Ready  Error  Retry    Ended    Errors
-----   ---------  --------  ----     ---------   ---------    -----  -----  -----  -----  ------   ------   ------
        Bus 1
        IOP 01
         (2666)

13:08   LOCALNET2    2048.0    78          842         333       10      8     22      6       4       11       20
13:23   LOCALNET2    2048.0    79          867         399       13      7     23      5       4       29       33
13:38   LOCALNET2    2048.0    82          997         522       23      8     32      7       7       52      123
```

Figure 103. Resource Interval Report: Communications Line Detail – FRLY

Communications Line Detail – ASYNC Sample

```
                                                    Resource Interval Report                                              04/19/90 13:58:59
                                                    Communication Line Detail                                                  Page     1
                                                    User-Selected Report Title

Member  . . . : MONDAY      Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA    System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = ASYNC (SORT BY INTERVAL)

            Bus/                                  Bytes         Bytes       Total       Pct PDUs
 Itv        IOP/       Line        Line       Transmitted     Received      PDUs       Received
 End        Line       Speed       Util         per Sec       per Sec     Received     in Error
-----    ----------  --------      ----     -----------     --------    --------     --------
        Bus 1
        IOP 01
         (6240)

13:08    STARTSTOP          0.3      18             343          433         343           03
13:23    STARTSTOP          0.3      23             343          433         343           00
```

Figure 104. Resource Interval Report: Communications Line Detail – ASYNC

Communications Line Detail – BSC Sample

```
                                                    Resource Interval Report                                              04/19/90 13:58:59
                                                    Communication Line Detail                                                   Page    1
                                                    User-Selected Report Title

Member  . . . : MONDAY      Model/Serial  . . :  B40/XX-XXXXX   Main storage . . :   16.0 M   Started  . . . . : 01/19/90 13:03:19
  Library . . : QPFRDATA    System name . . . :      SYS400     Version/Release  :   2/ 2.0   Stopped  . . . . : 01/19/90 14:57:50

 PROTOCOL = BSC (SORT BY INTERVAL)

                                                                   Pct Data                                   Pct Data
           Bus/                               Bytes      Total Data   Characters        Bytes    Total Data   Characters
 Itv       IOP/      Line       Line     Transmitted   Characters   Transmitted     Received   Characters    Received     Line
 End       Line      Speed      Util        per Sec    Transmitted    in Error       per Sec     Received     in Error    Errors
-----   ----------  --------    ----    -----------  -----------  -----------   ---------   ----------   ----------   ------
        Bus 1
        IOP 01
         (6240)

13:08   PNTTOPNT          4.8     30            444   12,345,444            09         104       34,211           01      383
13:23   PNTTOPNT          4.8     24            444   12,345,444            05         104       34,211           00      121
```

Figure 105. Resource Interval Report: Communications Line Detail - BSC

Communications Line Detail – ISDN Network Interface Sample

Resource Interval Report — 01/10/92 08:00:33
Communications Line Detail — Page 10
User-Selected Report Title

Member . . . : MONDAY   Model/Serial . . : D80/XX-XXXXX   Main storage . . : 160.0 M   Started . . . : 05/07/91 14:31:22
Library . . : QPFRDATA   System name . . . : SYS400   Version/Release . : 2/2.0   Stopped . . . : 05/07/91 16:26:10

PROTOCOL = ISDN NETWORK INTERFACE (SORT BY INTERVAL)

Itv End	Bus/ IOP/ Network Interface	Line Speed	--Outgoing--- ---Calls----- Total	Pct Retry	---Incoming--- ----Calls----- Total	Pct Reject	LAPD Total Frames Trnsmitd	LAPD Pct Frames Trnsmitd Again	LAPD Total Frames Recd	LAPD Pct Frames Recd in Error	Loss of Frame Alignment	Local End Code Violation	Collision Detect
	BUS 7 IOP 3 (2623)												
14:46	ISDNSS_A	16.3	42	28	15	26	256	60	233	36	452	97	112
14:46	ISDNSS_B	16.3	5	0	0	0	132	0	132	0	0	0	0
15:01	ISDNSS_A	16.3	74	74	33	99	187	13	382	48	135	107	91
15:01	ISDNSS_B	16.3	5	0	0	0	129	0	129	0	0	0	0

Figure 106. Resource Interval Report: Communications Line Detail - ISDN Network Interface

Communications Line Detail – NWI Maintenance Sample

Resource Interval Report — 01/10/92 08:00:33
Communications Line Detail — Page 13
User-Selected Report Title

Member . . . : MONDAY   Model/Serial . . : D80/XX-XXXXX   Main storage . . : 160.0 M   Started . . . : 05/07/91 14:31:22
Library . . : QPFRDATA   System name . . . : SYS400   Version/Release . : 2/2.0   Stopped . . . : 05/07/91 16:26:10

PROTOCOL = NWI MAINTENANCE CHANNEL (SORT BY INTERVAL)

Itv End	Bus/ IOP/ Network Interface	Line Speed	Percent Errored Seconds	Percent Severely Errored Seconds	-----Detected Access----- ----Transmission Error--- In	Out	Far End Code Violation
	BUS 7 IOP 3 (2623)						
14:46	ISDNSS_A	16.3	50	36	734	83	32
15:01	ISDNSS_A	16.3	6	24	32	14	52
15:16	ISDNSS_A	16.3	0	0	0	0	0

Figure 107. Resource Interval Report: Communications Line Detail - NWI Maintenance Channel

Communications Line Detail – IDLC Samples

```
                                               Resource Interval Report                                    01/10/92 08:00:33
                                              Communications Line Detail                                           Page  15
                                              User-Selected Report Title

 Member  . . . : MONDAY      Model/Serial  . . : D80/XX-XXXXX    Main storage  . . : 160.0 M     Started . . . : 05/07/91 14:31:22
   Library . . : QPFRDATA    System name . . . :     SYS400      Version/Release . :   2/2.0     Stopped . . . : 05/07/91 16:26:10

 PROTOCOL = IDLC (SORT BY INTERVAL)

         Bus/                          Transmit/            ---Frames----            ---Frames----
         IOP/                          Receive/   Bytes     -Transmitted-  Bytes     --Received---  Receive                     Short
  Itv  Network    Line       Line      Average    Trnsmitd            Pct  Recd                Pct  CRC       Aborts   Sequence  Frame
  End  Interface  Descriptn  Speed     Line Util  Per Sec   Total     Err  Per Sec   Total     Err  Errors    Recd     Error     Errors
 ----- ---------- ---------- -----   ---------  --------  --------- ---  -------   --------- ---  -------   -------  --------  -------
        BUS  7
        IOP   3
         (2623)

 14:33 ISDNSS_B   ISDNSS_B1  16.3    99/99/99      3,451        347  33    2,236         130  39      328        93        835      153
 14:33 ISDNSS_A   ISDNSS_A1  64.0    01/02/02        139        905   0      197         904   0        0         0          0        0
 14:36 ISDNSS_B   ISDNSS_B1  16.3    02/01/02         52        410   0       37         409   0        1         1          0        0
 14:36 ISDNSS_A   ISDNSS_A1  64.0    00/00/00         38        409   0       52         410   0        0         0          0        0
```

Figure 108. Resource Interval Report: Communications Line Detail – IDLC

```
                                               Resource Interval Report                                    01/10/92 08:00:33
                                              Communications Line Detail                                           Page  22
                                              User-Selected Report Title

 Member  . . . : MONDAY      Model/Serial  . . : D80/XX-XXXXX    Main storage  . . : 160.0 M     Started . . . : 05/07/91 14:31:22
   Library . . : QPFRDATA    System name . . . :     SYS400      Version/Release . :   2/2.0     Stopped . . . : 05/07/91 16:26:10

 PROTOCOL = IDLC (SORT BY INTERVAL)

         Bus/
         IOP/
  Itv  Network     Line
  End  Interface   Description  Channel
 ----- ----------  -----------  -------
        BUS  7
        IOP   3
         (2623)

 14:33 ISDNSS_B    ISDNSS_B1    B1
 14:33 ISDNSS_A    ISDNSS_A1    B1
 14:36 ISDNSS_B    ISDNSS_B1    B2
 14:36 ISDNSS_A    ISDNSS_A1    B2
```

Figure 109. Resource Interval Report: Communications Line Detail - IDLC

Communications IOP Utilizations – Sample

```
                                              Resource Interval Report                                    04/19/90 13:58:59
                                          Communications IOP Utilizations                                          Page    7
                                             User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

Bus    IOP ID/      Itv                     -- OPSTART Msg --     --- Bytes Transmitted ---     Restart    BNA      Avail Local
ID     Model        End      Utilization    Reverse    Normal         IOP            System      Queues   Received   Storage (K)
---    ----------   -----    -----------    -------    ------     ----------     ----------     -------   --------   -----------
  0     05 (6110)   09:19         1.6             0        95         13,786         79,560           0          0           160
                    09:34         2.0             0       201         17,488        247,280           0          0           160

  1     02 (6110)   09:19         3.1             0       469        173,922        262,933           0          0           687
                    09:34          .7             0        86          1,138         77,551           0          0           687
```

Figure 110. Resource Interval Report: Communications IOP Utilizations

Disk IOP Utilizations – Sample

```
                                                   Resource Interval Report                                      04/19/90 13:58:59
                                                    Disk IOP Utilizations                                                Page    8
                                                  User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

Bus      IOP ID/        Nbr      Itv                 ------ Disk I/O /Sec ------     --- KB per I/O ----
ID       Model          Arms     End       Util        Reads           Writes        Read        Write
---      ----------     ----     -----     ----      -----------     -----------     -------     -------
  0       01 (6110)        7     09:19       .7             .595            .685         6.9         4.9
                                 09:34       .6             .635            .326         4.3         4.4

  0       09 (6110)       16     09:19      2.2            1.878           1.637         9.5         5.8
                                 09:34      1.3             .915            .845        10.5         5.7
```

Figure 111. Resource Interval Report: Disk IOP Utilizations

Multifunction IOP Utilizations – Sample

```
                                                   Resource Interval Report                                      04/19/90 13:58:59
                                                Multifunction IOP Utilizations                                           Page    9
                                                  User-Selected Report Title

Member  . . . : WASCOMTEST   Model/Serial  . . : B60/10-12883    Main storage  . . : 96.0 M      Started . . . : 01/18/90 09:04:26
  Library . . : QPFRJAG      System name . . . :     TEST#366    Version/Release . :  2/2.0      Stopped . . . : 01/18/90 09:34:00

Bus      IOP ID/         Itv
ID       Model           End        Utilization
---      ----------      -----      -----------
  0       05 (6110)      09:19              1.6
                         09:34              2.0

  1       02 (6110)      09:19              3.1
                         09:34               .7
```

Figure 112. Resource Interval Report: Multifunction IOP Utilizations

Local Workstation IOP Utilizations – Sample

```
                                                   Resource Interval Report                                      01/10/92 06:46:23
                                              Local Work Station IOP Utilization                                         Page  61
                                                  User-Selected Report Title

Member  . . . : MONDAY       Model/Serial  . . : B70/XX-XXXXX    Main storage  . . : 96.0 M      Started . . . : 11/08/91 14:16:58
  Library . . : QPFRDATA     System name . . . :       SYS400    Version/Release . :  2/2.0      Stopped . . . : 11/08/91 16:31:41

Bus    IOP Id/     Work Station   Itv              Active                                                                    Rsp
Id     (Model)     Controller     End       Util   Wrk Stn     0.0- 1.0     1.0- 2.0     2.0- 4.0     4.0- 8.0     > 8.0    Time
---    ---------   ------------   -----     ----   -------    ---------    ---------    ---------    ---------    -------   -----
  0    03 (6040)   CTL01          14:21      6.9         1            0            1            0            0          0     2.0
                                  14:26      7.2         3           12            5            2            2          2     1.1

  0    04 (6040)   CTL02          14:21      4.7         1           11            1            1            0          0     2.0
                                  14:26      5.4         0            0            0            0            0          0      .0
                                                              ---------    ---------    ---------    ---------    -------   -----
Total Responses:                                                    365           30           22            9         34     1.0
```

Figure 113. Resource Interval Report: Local Workstation IOP Utilizations Se ction

Remote Workstation Response Times – Sample

```
                                                  Resource Interval Report                                    04/05/95 14:53:50
                                             Remote Work Station Response Times                                        Page  14
                                                           5494 RT
Member  . . . : RWSRRSPTIM   Model/Serial  . . : D60/10-15181    Main storage  . . : 80.0 M      Started . . . : 04/05/95 14:27:51
  Library . . : COOK         System name . . . :     RCHASM01    Version/Release . :  3/1.0      Stopped . . . : 04/05/95 14:37:50

Bus    IOP Id/     Work Station   Itv              Active                                                                  Rsp
Id     (Model)     Controller     End              Wrk Stn     0.0- 1.0     1.0- 2.0     2.0- 4.0     4.0- 8.0     > 8.0   Time
---   ---------    ------------  -----             -------    ---------    ---------    ---------    ---------   -------  -----
  0   05 (6130)    RWS5494C01     14:32                  1            2            5            4            2         1    3.4
                                  14:37                  1            0            1            0            3         1    5.2
                                                             ----------   ----------   ----------   ---------- ----------  -----
Total Responses:                                                      2            6            4            5          2    3.8

Bus ID                  -- Bus identification number
IOP ID/(Model)          -- Input/Output processor identification number and
                           model number of the attached device
Work Station Controller -- Work station controller description name
Itv End                 -- Interval end time (hour and minute)
Active Wrk Stn          -- Number of work stations with activity
 0.0- 1.0               -- Number of response times between  0.0 and  1.0 seconds
 1.0- 2.0               -- Number of response times between  1.0 and  2.0 seconds
 2.0- 4.0               -- Number of response times between  2.0 and  4.0 seconds
 4.0- 8.0               -- Number of response times between  4.0 and  8.0 seconds
   > 8.0                -- Number of response times >  8.0 seconds
Rsp Time                -- Average external response time (in seconds) for
                           work stations on this controller
```

V3R1 example

Figure 114. Resource Interval Report: Remote Workstation Response Times

B.33 Lock Report

```
 11/18/88  13:45:40                         Seize/Lock Wait Statistics by Time of Day                                  Page   1
  TOD of   Length                                                         Object                                       Record
   Wait    of Wait L Requestor's Job Name         Holder's Job Name         Type    Object Name                        Number
 --------  ------- - ---------------------------- ---------------------------- ------ ------------------------------ ----------
 13.33.15     2033 L SCPF       QSYS       000000 WKS12502   ROBERT     159708 MSGQ   QHST       QSYS
 13.33.25     3372   WKS04001   JACKIE     159505 WKS20601   NANCY      159599 MBR    INVMAST    PROD1      INVMAST
 13.33.28     2344 L WKS13300   MASTER     159706 SCPF       QSYS       000000 MSGQ   QHST       QSYS
 13.33.36    10871   QSYSARB    QSYS       159052 WKS12202   JIMST      159627 CUD    C00B99
 13.33.36     2187   WKS03000   TERRI      159701 WKS12003   KAREN      159524 MBR    ORDER01    PROD1      ORDER01
 13.33.40     2400   AUT0001    MASTER     159546 WKS01009   WENDY      159387 MBR    CUSTMAST   PROD1      CUSTMAST
 13.33.52     3881   WKS04001   JACKIE     159505 WKS12102   JIM        159392 MBR    INVMAST    PROD1      INVMAST
 13.33.52     4386   WKS05001   HARRY      159196 WKS12020   JIM        159392 MBR    INVMAST    PROD1      INVMAST
 13.33.00     2289   WKS05001   HARRY      159196 WKS04000   NANCY      159527 MBR    INVMAST    PROD1      INVMAST
 Member SORTED          Library QPFRDATA       Period from 13.33.00 through 13.34.00      2,000 ms minimum wait
```

Figure 115. Example of a Detail Listing

```
11/18/87 13:45:40                              Seize/Lock Wait Statistics Summary                                       Page   5
                                                         Locks                          Seizes
Requestor's Job Name                               Count      Avg Length          Count       Avg Length
----------------------------                   ------------- -------------     ------------- -------------
AUT0001    MASTER     159546                                                               1         2,400
WKS13300   MASTER     159706                               1         2,344
QSYSARB    QSYS       159052                                                               1        10,871
WKS03000   TERRI      159701                                                               1         2,187
WKS04001   JACKIE     159505                                                               2         3,627
WKS05001   HARRY      159196                                                               2         3,338
SCPF       QSYS       000000                               1         2,033
Member SORTED         Library QPFRDATA       Period from 13.33.00 through 13.34.00      2,000 ms minimum wait
```

Figure 116. Example of Summary by Requesting Job

B.34 Batch Job Trace Report

```
                                          Batch Job Trace Report                                     06/13/90 14:01:46
                                               Job Summary                                                   Page    1
                                       Sample Batch Job Trace Report

Member  . . . :   GOODSTUF   Model/Serial  . . :  B60/10-15018   Main storage . . :    96.0 M  Started . . . . . :   6/07/90 08:09:22
  Library . . :   QPFRDATA   System name . . . :      RCH38366   Version/Release  :    2/ 2.0  Stopped . . . . . :   6/07/90 08:41:55

                                                                     --- Physical --     Seize       EAO     --- State ---
   Job          User          Job               -- Job --   Number    CPU    -- I/O Count --    and Lock     Excp    -Transitions-
   Name         Name          Number     Pool   Type  Pty   Traces    Util    Sync      Async    Conflicts   /1000     A-A     A-I
----------   ----------    ------    ----   ----  ---   ------    ----   ------    ------    ---------   -----   -----   -----
C12TS1       CLASS23       005276     02      B    50      395    10.6    1,155       350           95     1.5      27      11
C13T         QSECOFR       005277     02      B    50      428    16.6    1,137       523          123      .5       0       0
C13TS1       CLASS33       005236     02      B    40        1      .0      103        23        1,446      .0     112      89
C14T         QSECOFR       005275     03      B     0       16      .4      518       351           87      .1       5      37
C14TS1       CLASS43       005230     02      B    40        1      .0       49         8          267      .0      33       6
```

Figure 117. Job Summary

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 will determine when a customer can benefit from the use of 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 will be of particular interest:

- 226NC
- 130NC

Many informal sources of information, like 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.

The following publication is an ″as is″ document that can be obtained only by systems engineers and business partners. The information in the document can be given to customers as it is continually presented to many customer groups. This document contains details on performance components and the impact of specific resource utilizations on performance. Much of the information in this document is based on the following workshop notes:

- *AS/400 Performance Analysis Workshop Presentation Notes*, available from VM TOOLS with the following VM request: REQUEST PERFV2R2 PACKAGE FROM PAHOEHOE AT RCHVMW (your name

The *AS/400 Performance Capabilities Reference* - manual 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 termial:

```
REQUEST V3R1 FROM FIELDSIT AT RCHCMW2 (your name
```

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 personal computer.

QSIZE400 supports manual entry of 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 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 etc.

Once the planned workload is developed, using measured profiles or predefined profiles, QSIZE400 can be used to estimate the capacity of the AS/400 that would meet 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. See C.6, "Work Station Function (WSF) Measured Profile" on page 418 for a tool that allows you to include WSF PC Support activity.

C.3 Analyzing System/36 Performance before Migration

The AS/400 tools disk (AS4TOOLS) contains the System/36 tool TRACE36, which can be run on the System/36 before the applications are migrated to the AS/400. This tool and the associated documentation can help identify System/36 programs and procedures that do frequent file opens and closes and file creations and deletions.

File opens and closes and file creations and deletions are CPU and disk intensive operations on the AS/400. Because there are enhancements in these areas for each AS/400 release, including the Version 2 improved reuse of System/36 opened file control blocks, use of TRACE36 is strongly recommended.

To order the tool, use the AS4TOOLS tools disk procedures on your IBM VM system.

Notice

This tool is no longer recommended. You should use the equivalent ″Trace36″ support that runs on the System/36 before migration to AS/400. This ″Trace36″ support is available on the System/36 with the Value Added Software Package support.Refer to *Using Value-Added Software Package - 9402 Model 236*, SC21-8368, for more information.

C.4 AS/400 Storage Management Trace Tool

After using the Performance Tools/400 functions including the Analyze PAG, Sampled Address Monitor, and Disk Collection support and the TPST PRPQ, you may still not be able to determine what user or IBM processing is the cause for high CPU utilization, high disk I/O operations or what objects are being referenced ″excessively.″ When you have exhausted the usage of formally available performance tools, you should consider using the SMTRACE tool set that can be downloaded to a customer system as a PTF and run on the AS/400 with the performance problems.

C.4.1 General Description

The set of libraries, programs, commands and queries collectively called **SMTRACE** comprise a tool which performs tracing and analysis of various detailed, low-level functions performed on an AS/400. The trace point data included in the tool are:

- Physical disk I/Os
- Logical disk I/Os, including page faults and explicit paging requests
- Auxiliary storage management operations: create, extend, truncate and destroy segments/objects (physical disk space)
- ″Complex″ MI instruction starts
- External MI program invocations
- CPU ″profiling″ (that is, how busy the processor(s) is(are) every 100 milliseconds)
- Low level CPU usage within processes (jobs)

A wide variety of problem analysis/investigation areas can be examined with this tool. Areas included are:

- Faulting and paging analysis
  - What objects are faulting?
  - What objects are paging overall?
  - What objects are paging in pool n?
  - What objects are paging on disk unit n?
  - Identifying possible candidates for SETOBJACC command
  - What object *types* are paging overall?
  - What is the overall CPU utilization on a microscopic scale?
  - What jobs/programs are using ″excessive″ CPU?
- Analysis of average disk I/O transfer sizes (possibly identifying poor data base blocking)
- %-free-disk-space decreasing too quickly
- %-temp-addresses-used increasing too quickly
- %-perm-addresses-used increasing too quickly

The general scenario is to use the SMTBCH command to start the SMTRACE with the appropriate trace types and time limit and then run the various queries to analyze the collected data as described in the SMTRACE User′s Guide, an internally available document.

Notice

Usage of the SMTRACE tool requires use of an SMTRACE User′s Guide, at least an intermediate level understanding of AS/400 implementation and completion of detailed performance problem analysis. The SMTRACE User′s Guide contains problem analysis examples. The associated query output may identify customer generated causes, but may also point to IBM Software causes. Be prepared to request laboratory assistance if the cause appears to be IBM software.

C.4.2 How to Receive SMTRACE Code and Documentation

The SMTRACE User′s Guide can be obtained by sending a PROFS note to RCHVMV2(KURTZ)

You may download the SMTRACE tool set as a PTF according to the appropriate target system release level. The following set of PTFs is valid as of April, 1995:

- **V2R2**
  - MA06306 - library QDEVELOP
- **V2R3**
  - MA08050 - library QDEVELOP
  - MA08051 - library SMTRACE
  - MF08321 - SNDPTFORD PTF, which correctly reports library name for physical and logical files
- **V3R0.5**
  - MA09255 - library QDEVELOP
  - MA09256 - library SMTRACE
  - MF08322 - SNDPTFORD PTF, which correctly reports library name for physical and logical files
- **V3R1**
  - MA09247 - library QDEVELOP
  - MA09248 - library SMTRACE
  - MA10191 - additional RSTOBJ items for the SMTRACE library
  - MF08323 - SNDPTFORD PTF, which correctly reports library name for physical and logical files

Note: Contact the RCHVMV2(KURTZ) for a V2R2 version of the query library SMTRACE.

C.5 An AS/400 S/36 Environment Performance Analysis Tool

C.5.1 TRACE JOB TOOL (36E and AS/400)

This tool was created to analyze migrated System/36 applications that were achieving poor performance on the AS/400.

The tool analyzes AS/400 System/36 Environment (36E) performance or an AS/400 job. The tool was intended for 36E job analysis and was made available before Version 2 performance enhancements to AS/400 Submit Job and 36E create and delete file. Its use should still be considered if 36E jobs are experiencing poor performance.

Notice

This tool remains on the IBM VM AS4TOOLS disk but is no longer recommended. You should analyze the application *before* migraton from the System/36. Use TRACE36 on the System/36 as described above.

Analysis of many 36E applications shows that functions such as file open/close, file create/delete, and submit (evoke) take up a large part of the disk and CPU resources. One part of the solution is to find where this is occurring and then make application changes to minimize their occurrence.

This tool that can help you find:

- Which high use AS/400 functions are being used
- What procedures and/or programs need to be changed to reduce resource usage.

TRACE36E is not an AS/400 licensed program or a programming RPQ. It is available through the Rochester Competency Center on an ″as is″ basis. See C.5.2, “Summary” on page 418.

The overall procedure for using TRACE36E is:

1. Install the library containing the new trace job report commands.
2. Identify a job (or set of jobs) that are not performing as well as expected in 36E.
3. Obtain a job trace over the selected job.

   You can use either the OS/400 series of commands STRSRVJOB, TRCJOB SET(*ON), TRCJOB SET(*OFF), ENDSRVJOB or the Performance Tools series of commands (or menu options): STRJOBTRC, ENDJOBTRC to obtain a job trace.
4. When complete, use the TRACE36 library CRTTRCFUNC and PRTTRCFUNC commands to print output showing the frequency, type, and program or procedure using the ″high cost″ functions (open/close, submit job, etc.)

   Use the command CRTTRCFUNC to input the necessary details. CRTTRCFUNC will create the necessary file and optionally print the reports. You can print either the detail or summary reports or both. Also one of the parameters is to specify whether you want the procedure specific (S36E) or program specific (AS/400) reports.

5. Determine if the problem is as anticipated (open/close, etc.) and if so, what procedures/programs are using the high cost functions.

Once you have generated the files using CRTTRCFUNC then you can use the PRTTRCFUNC command to print the reports any number of times.

```
         Create Trace Function File   (CRTTRCFUNC)

                                                            ┌────────────────┐
 Job Trace File Name  . . . . . .   QAPTTRCJ      Name      │File, Library,  │
                                                            │and Member      │
                                                            │from TRCJOB OFF │
 Library   Name . . . . . . . . .   QPFRDATA      Name      └────────────────┘

 Member Name for Trace File . . .   *FIRST        Character value

 Member Name for Output File  . .   *FIRST        Character value

 Text For Output Member . . . . .   ______________________________

 S=Summary , D=Detail , B=Both  .   _             B, S, D, ' '

 P=By Program, R=By Procedure   .   P             P, R


            ┌──────────────────────────────────────────────────┐
            │USED TO PRINT SUMMARY OR DETAIL REPORTS BY PROGRAM│
            │OR PROCEDURE.                                     │
            │                                                  │
            │BUILDS INTERMEDIATE DATA BASE FILES FOR INPUT TO  │
            │'PRTTRCFUNC' FOR ADDITIONAL REPORT RUNS           │
            └──────────────────────────────────────────────────┘
```

The command parameters are:

FILE Trace file name from which the report is generated

LIBR Library in which the above file resides

TRCMBR Member to be processed in the above file

OUTMBR Member of TRCPF1 in which the result is stored

TEXT Text for the OUTMBR

REPORT Type of report (detail or summary) to be printed

PGM Report to be sequenced by program or procedure, (P = Program and R = Procedure)

```
                    Print Trace Function File  (PRTTRCFUNC)

 Member from which to print . . .   *FIRST        Character value

 S=Summary , D=Detail , B=Both  .   B             B, S, D

 P=By Program, R=By Procedure   .   P             P, R

           ┌──────────────────────────────────────────────┐
           │USED TO PRINT SUMMARY OR DETAIL REPORTS BY PROGRAM│
           │OR PROCEDURE.                                    │
           │_________________________________________________│
```

The command parameters are:

OUTMBR Member of TRCPF1 in which the result is stored

REPORT Type of report (detail or summary) to be printed

PGM Report to be sequenced by program or procedure

```
            SUMMARY OF CPU EXPENSIVE FUNCTIONS
            __________________________________

Function  Description          Count     ┌───────────────────────────────┐
_____________________          _____     │SHOWS THE TOTAL COUNT OF THE   │
                                         │HIGH USE FUNCTIONS AS WELL AS  │
CREATING A MEMBER                  1     │THE USE OF PREFERRED FUNCTIONS │
                                         │(SHARED DATA BASE OPEN AND     │
FULL CLOSING OF A FILE             5     │CLOSE).                        │
                                         └───────────────────────────────┘
FULL OPENING OF A FILE             8

SHARED CLOSING OF A FILE           4

SHARED OPENING OF A FILE          18


TOTAL COUNT     36
___________
```

```
OPTION SELECTED) THE USE OF

HIGH COST FUNCTIONS.
```

Other Sample Output

```
                SUMMARY OF CPU EXPENSIVE FUNCTIONS
               ___________________________________

Program         Function  Description          Count
_______         _____________________          _____

CAE002          FULL OPENING OF A FILE             2
                SHARED OPENING OF A FILE          14

TOTAL COUNT FOR PROGRAM
_______________________   CAE002       16


MAUR01          FULL OPENING OF A FILE             1
                SHARED CLOSING OF A FILE           4
                SHARED OPENING OF A FILE           4

TOTAL COUNT FOR PROGRAM
_______________________   MAUR01        9



xxxxxxxx        CREATING A MEMBER                  1
                FULL CLOSING OF A FILE             5
                FULL OPENING OF A FILE             5

TOTAL COUNT FOR PROGRAM
_______________________   xxxxxxxx     11


TOTAL COUNT
___________     36
```

```
SHOWS BY PROGRAM (OR PROCEDURE DEPENDING
ON THE COMMAND OPTION SELECTED) THE USE
OF HIGH COST FUNCTIONS.  ALSO GIVES A
TOTAL COUNT FOR EACH PROGRAM OR
PROCEDURE.

NOTE:  xxxxxxx INDICATES THAT THE
PROGRAM NAME COULDN'T BE DETERMINED
SINCE IT WAS ACTIVE AT THE START OF
TRACING
```

```
Program 1 Function Description     File Name Library Member Name Program 2 ...

CAE002    FULL OPENING OF A FILE   CAE002FM  CABLLRO DSP10       DRIVER
                                   DEBUG     QS36F   M891221     DRIVER

          SHARED OPENING OF A FILE CB.CACMT  QS36F   M891212     DRIVER
                                   CB.CADCP  QS36F   M891212     DRIVER
                                   CB.CAHCP  QS36F   M891212     DRIVER
                                   CB.CAPRT  QS36F   M891212     DRIVER
                                       .        .        .           .

Program totals                     CB.SYCTL  QS36F   M880210     DRIVER

COUNT 16                           CB.SYCTL  QS36F   M880210     DRIVER

XXXXXXXX CREATING A MEMBER

         FULL CLOSING OF A FILE    CMPMNA    CABLLRO DSP10
                                   CMPMNA    CABLLRO DSP10
                                   MANMNA    CABLLRO DSP10
                                   MAURO1FM  CABLLRO YJ
                                   QDPTDSP   QSYS    DSP10

         FULL OPENING OF A FILE    ##FCPF    QSSP    DSP10
                                   CMPMNA    CABLLRO DSP10
                                   CMPMNA    CABLLRO DSP10
Program totals                     MANMNA    CABLLRO DSP10

COUNT 11                           QDPTDSP   QSYS    DSP10

Total of functions

COUNT 27
```

```
┌─────────────────────────────────────────────────────┐
│SHOWS BY PROGRAM (OR PROCEDURE DEPENDING ON THE      │
│COMMAND OPTION SELECTED) THE USE OF HIGH COST        │
│FUNCTIONS.  GIVES A TOTAL COUNT FOR EACH PROGRAM OR  │
│PROCEDURE AND WHICH FILE IS INVOLVED.                │
└─────────────────────────────────────────────────────┘
```

```
Description               File Name   Member Name  Library   Pgm1     Pgm2...

CREATING A MEMBER                                   XXXXXXXX

Function Total
COUNT 1
-------

FULL CLOSING OF A FILE   CMPMNA    DSP10    CABLLRO  XXXXXXXX
                         CMPMNA    DSP10    CABLLRO  XXXXXXXX
                            .        .         .          .
                         QDPTDSP   DSP10    QSYS     XXXXXXXX
Function Total
COUNT 5
-------

FULL OPENING OF A FILE   ##FCPF    DSP10    QSSP     XXXXXXXX
                         CAE002FM  DSP10    CABLLRO  CAE002    DRIVER
                         CMPMNA    DSP10    CABLLRO  XXXXXXXX
                         CMPMNA    DSP10    CABLLRO  XXXXXXXX
                            .        .         .          .
                         QDPTDSP   DSP10    QSYS     XXXXXXXX
Function Total
COUNT 8
-------

SHARED CLOSING OF A FILE CB.MACTY  M880705  QS36F    MAUR01    CAE002  DRIVER
                            .        .         .          .
                         MAUR01FM  YJ       CABLLRO  MAUR01    CAE002  DRIVER
Function Total
COUNT 4
-------                 ┌────────────────────────────────────────────────────┐
                        │SHOWS THE DETAIL BY FUNCTION GIVING THE FILE,       │
Total of all functions  │LIBRARY, AND PROGRAM/PROCEDURE NAME FOR EACH USE.   │
                        │NOTE THAT DUPLICATE FILE AND PROGRAM NAMES INDICATE │
COUNT 18                │MULTIPLE USE OF THE FUNCTION.                       │
-------                 └────────────────────────────────────────────────────┘
```

C.5.2 Summary

TRACE36E is available for use by either IBM or non-IBM users on an ″as is″ basis. No warranty as to its accuracy or continued support is made or implied.

To order the tool, use the AS4TOOLS tools disk procedures on your IBM VM system.

This tool requires the AS/400 tools disk Convert Library (CVTLIBRM) package so you can download the package from VM to 3270 PC to AS/400.

C.6 Work Station Function (WSF) Measured Profile

This tool was created mainly to address the limitations of MDLSYS to model WSF workloads. Use the tool described herein only if you are on a release prior to V2R2 and want to use MDLSYS to model WSF workloads. The BEST/1 Capacity Planning Tool has the capability to model natively WSF workloads and other workloads unsupported by MDLSYS. In this case we highly recommend using

BEST/1. You may still want to use this tool, if for some reason, you are using measured profiles generated with PRTSYSRPT. This is the case if you are using APMT or QuickSizer/400 with measured profiles.

The Performance Tools Measured Profile function (PRTSYSRPT command) does not support modeling Personal Computer (PC)-based Work Station Function jobs, even though these jobs are measured and included in the various Performance Tools printed reports.

An ″as is″ tool has been developed that addresses a major portion of this requirement. This tool changes specific data values in the Performance Monitor database files so that the measured profile, created with PRTSYSRPT, interprets the WSF jobs as if they were local twinaxial-attached workstations.

This technique provides the most accurate WSF measured profiles when the measured Performance Monitor data includes only WSF jobs and no AS/400 PC Support Shared Folder jobs. This technique has been shown to provide good measured profile information for CPU, memory, and disk requirements. However, since WSF jobs are treated as local workstation controller jobs, any tool recommendations that use PRTSYSRPT measured profiles regarding workstation controllers and WSC-attached devices must be considered inaccurate if this ″as is″ tool has been used to change the files.

The onus is on the user of this tool to determine that any modeled predictions are reasonable. The tool runs on the AS/400.

To order the tool, use the AS4TOOLS tools disk procedures on your IBM VM system.

Considerations and restrictions are discussed below.

C.6.1 Considerations

Warning

Performance data that has been modified with MDLWSF, might produce erroneous results using other Performance Tools functions besides the measured profile report and files. In particular, if you want to use the same data with V2R2 Capacity Planning, save your data before using MDLWSF and always use your original data to create your models.

The modifications to the QAPMSYS file cause statistics for Work Station Function (WSF) to be combined with normal interactive statistics. File Transfer and Shared Folder statistics are treated as a mixture of batch and system statistics. The subsequent System and Component reports should be ignored as they may contain meaningless data (depends upon original job mix). However, the measured profile will correctly combine interactive and WSF transaction data to produce a profile, subject to the following restrictions.

C.6.2 Restrictions

- This technique should be used only for measured profiles with respect to CPU, main storage and disk utilization.
- This technique can result in measured profiles that have too many local workstations for the number of local workstation controllers.
- Be aware that file transfer activity is split between system and batch job. activity including WSF and file transfer (named #Bxxxx, #3xxxx and #5xxxx), but not the batch WSF server jobs. You should ensure that file transfer activity is minimal or not present in your original data.
- Be aware that shared folder activity is split between system and batch jobs. You should ensure that shared folder activity is minimal or not present in your original data. Look at the Component report or run a query over the QAPMJOBS file member to check CPU utilization of the aforementioned jobs. If the load is significant, you should look for another set of data where the load is less.
- This technique **has not** been validated by approved benchmark measurements. Such measurements are necessary to ensure that workloads will behave as expected at various degrees of throughput, for various system configurations. However, several customer configurations have been tested and proven to give reasonable performance prediction information for CPU, memory, and disk utilization.

Cross-release implementation:

```
PRTSYSRPT  |will work on data from   | <-- after CVTPFRDTA to appropriate
release    | V1R1|V1R2|V1R3|V2R1|V2R2|       release and MDLWSF used
V1R1       |  nt | na | na | na | na |  Y = Yes, tested
V1R2       |  nt | nt | na | na | na |  nt = not tested, may work
V1R3       |  nt | nt | nt | na | na |  na = not applicable
V2R1       |  nt | nt |  Y |  Y | na |
V2R2       |  nt | nt | nt | nt |  Y |
```

C.7 AS/400 IBMLIB

C.7.1 General Instructions

The library IBMLIB was originally designed to be able to analyze the performance impact of going from Version 1 Release 2 to Release 3. The latest version provides this function and other functions including graphics, which are useful on Version 2, but there is no support for analyzing performance impact of going from Version 1 Release 3 to Version 2 Release 1.

The graphics support in IBMLIB can be used as an alternative or supplement to the graphics support in the Performance Tools licensed program 5738-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.

C.7.2 New News / Modifications

This is an introduction to Version 1.1 of IBMLIB tools. The first version was distributed to some US system engineers via tape and contained libraries IBMLIB and IBMEXMPL01. Version 1.1 consists of just library IBMLIB, which can be obtained electronically through a **VM PROFS REQUEST**.

- REQUEST xxxxxxxx FROM CRAVENS AT RCHVMP (your-name

If xxxxxxxx = IBMLIBAS, an AS/400 SAVF of library IBMLIB will be sent. If you have an AS/400 hooked up to VM, then this would be the most convenient package to ask for. If xxxxxxxx = IBMLIBPC, a library file in the AS/400 tools disk format will be sent. This means you will need to use the AS400TLS disk CVTLIBRM function to download from the S/370 through a PC to an AS/400.

IBMLIB Version 1.1 includes two new commands for performing a more detailed analysis of authority counts. The AUTHTRC command starts an authority management trace and the AUTHPRT command prints a summary of the objects and jobs with private authority counts.

Appendix A, ″Guidelines for Interpreting Performance Reports,″ includes a table showing approximate CPU impact of authority lookups. The Performance Tools Component Report lists ″Aut Lookup″ exceptions per second number. Use the table in Appendix A with this ″Aut Lookup″ value to determine if the CPU impact is worth investigating further. The EXCEPTION command in IBMLIB does this for you. Note that the Performance Tools Advisor function will produce a ″conclusion″ if the guideline is exceeded. There is no system or Performance Tools function to identify the objects with excessive authority lookups. AUTHTRC and AUTHPRT commands in IBMLIB can help identify the objects with excessive private authority counts.

IBMLIB Version 1.1 differs from Version 1.0 in two other ways:

1. The source code for the RPG programs has been eliminated from the SOURCE file. This was done to reduce the electronic traffic as well as protect some IBM resource.
2. The major programs have had observable information removed, again to reduce the electronic traffic by reducing the object size of the larger compiled programs.

Study the new commands AUTHTRC and AUTHPRT for more details. They are outlined later in this section.

Be advised that the programs provided are experimental. The graphing programs reflect measured data but all combinations of parameter options have not been tested. It is recommended that only the command defaults be used.

C.7.3 Detailed Steps for Analyzing Data

C.7.3.1 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 then 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.

Just 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 is an example of using the GPHSMRY command:

```
GPHSMRY INPLIB(IBMR2DTA) CUSNAM('customer name')
```

See C.7.4, "GPHSMRY Command Details" on page 422 for a full description of this command and all its parameters.

C.7.3.2 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 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 then rerunning the GPHSMRY command in Step 1, or simply use date/time selection when running additional graphing analysis.

The following is an example of using the GPHCMD command:

```
GPHCMD  INPLIB(IBMR2DTA)
```

See C.7.5, "GPHCMD Command Details" on page 423 in this publication for a full description of this command and all its parameters. Note that the GPHCMD command can also be used to create a series of more detailed analysis graphs if required.

C.7.4 GPHSMRY Command Details

The GPHSMRY command is used to create a summary file (GPHSMRYP) that is used by the GPHCMD and PRTCMD commands. 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.

```
The GPHSMRY command and all its parameters are described below:

          command  parameter  default  other options
          -------  ---------  -------  -------------
          GPHSMRY  INPLIB              library-name
                   CUSNAM     *NONE    customer-name
                   OUTLIB     *INPLIB  library-name
                   PTYRNG1    '00-09'
```

```
PTYRNG2    '10-19'
PTYRNG3    '20-20'
PTYRNG4    '21-49'
PTYRNG5    '50-99'
MBROPTN    *REPLACE *ADD
```

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. It's easier to keep track of things by just using the default.

Parameters PTYRNG1 - PTYRNG5 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. This change is not necessary for the CPU analysis report but might make the graphs easier to relate to interactive CPU usage and response time analysis.

The MBROPTN parameter default *REPLACE allows for a new summary graph file member to be created or replaced. The *ADD option will add new records to existing records in a member. If you use *ADD then you must use the customer description to select the records you want. Stick with the default.

C.7.5 GPHCMD Command Details

The GPHCMD command and all its parameters are described below:

```
command  parameter  default  other options
-------  ---------  -------  -------------
GPHCMD   INPLIB              library-name
         CUSNAM     *ALL     customer-name
         BEGDAT     *FIRST   6 digits (YYMMDD)
         BEGTIM     *FIRST   6 digits (HHMMSS)
         ENDDAT     *LAST    6 digits (YYMMDD)
         ENDTIM     *LAST    6 digits (HHMMSS)
         GRIDYN     *YES     *NO
         FORMAT     A        B
         SLCTXX     *HRS24   *SELECT
         TRNSECB    0.1      number
```

The BEGDAT, BEGTIM, ENDDAT, ENDTIM parameters allow date and time selection of data. Just remember that year is first on the date parameters.

The GRIDYN parameter allows for grid lines on the Y axis.

The FORMAT parameter has only two choices: A allows for just CPU analysis over time and will put up to four days on one graph page. B will cause a series of five detailed graph pages to be produced for each day, each page with up to four graphs per page. Stick with the A default for an overview and then be selective by day and time for the detail B option.

The SLCTXX 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.

The TRNSECB 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.

Note that when you take FORMAT option B to get some detailed graphs, you could see the message "Graphic messages signalled during processing." This is normal as some of the data may cause graph points that exceed the Y axis ranges specified.

C.7.6 AUTHTRC Command Details

The AUTHTRC command utilizes the TRCINT and DMPTRC commands to start and dump an authority management trace for a user-specified measurement period.

The AUTHTRC command and all its parameters are described below:

```
command  parameter  default   other options
-------  ---------  --------  -------------
AUTHTRC  INPLIB               library-name
         INPMBR     AUTHTRC#  8 character prefix
         RUMTM      60        number of seconds
         DLYTM      900       number of seconds
         NUMTI      1         number of trace intervals
```

The INPLIB parameter names the library to store the trace data file QAPMDMPT which will be created automatically if not present. You'll 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 will be created. The prefix will be 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.7.7 AUTHPRT Command Details

The AUTHPRT command prints a summary report showing private authority counts by object type and by job.

The AUTHPRT command and all its parameters are described below:

command	parameter	default	other options
AUTHPRT	INPLIB		library-name
	CUSNAM	*NONE	customer-name/description
	INPMBR	*ALL	Name, generic*, *ALL

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 will be included in the report.

C.8 BatchSizer Overview

BatchSizer is a Lotus 1-2-3**-based spreadsheet template that contains analytical equations to represent AS/400 behavior for batch and interactive workloads. Its primary use is for batch analysis, but interactive analysis is also available through a spreadsheet version of the Quicksizer (QSIZE400). This is included to facilitate combinations of batch and interactive workload analysis.

The spreadsheet can be used to make projections of batch workloads with or without interactive workloads on various AS/400 models while varying a number of system parameters (for example, checksum and mirrored protection) and batch characteristics.

This tool has received several program fixes since its original availability and recent uses have proven reasonably accurate. However, extensive feedback is not yet available. So caution is urged. Input from actual measurement data on a system with little or no additional job activity is strongly recommended.

The development laboratory is working on a batch capacity planning tool for future support. As of publication of this book, no product announcement has been made.

C.8.1 Batch Workload Definition

Two types of batch workload definition are provided:

- Measured - data can be input from an actual batch run.
- Generic - some simple profiles can be built.

C.8.2 Interactive Workload Definition

Two types of interactive workload definition are also provided:

- Manual - a simple interactive workload profile can be input.
- Quicksizer - you can use a spreadsheet version of Quicksizer to define an interactive environment made up from a mixture of predefined workloads.

C.8.3 Modeling Options

Three main variations of batch and interactive combinations are supported:

1. Dedicated batch with batch definition from either ″Measured″ or ″Generic.″
2. Batch with interactive at higher priority. Batch can be either measured or generic, and interactive can be either manual or Quicksizer origin.
3. Batch with interactive at equal priority. Batch can be either measured or generic, and interactive can be either from manual input or Quicksizer origin.

C.8.4 Output

BatchSizer output can either be viewed or graphed on the PC terminal. Several different sets of data are available for viewing, graphing, printing and plotting. Profiles can also be stored for subsequent retrieval.

C.8.5 How to Receive BatchSizer

The spreadsheet is available as part of the QBATCH package on the PCTOOLS repository. This package includes a user′s guide.

Note that LOTUS 1-2-3 Release 3 or Release 3.1 is required.

A detailed comparison of the functions of the two products is available in Appendix G of *Performance Tools Guide*.

C.9 Batch400 Overview

Batch400 is a V2R3 internal tool for ″batch window (job run time) analysis.″ It uses data collected by the OS/400 Performance Monitor.

C.9.1 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 can then be 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.

During 1994 ITSO residents used this tool and found it needed significant documentation updates to improve usability.

C.9.2 How to Receive Batch400

You can obtain Batch400 by entering the following on a VM terminal session:

```
REQUEST BATCH400 FROM DANFRIZ AT RCHVMW2
```

C.10 AS/400 IMAGESIZER Overview

C.10.1 General Description and Output

AS/400 ImageSizer is a set of tools used to size an AS/400 that will use the ImagePlus Workfolder Application Facility/400 (WAF).

ImageSizer is a LOTUS 1-2-3* spreadsheet template that does capacity planning for the WAF application. There is an ImageSizer version for each WAF supported release (as of April 1995 - V2R2 and V2R3). Non-image workloads may also be included.

Output includes:

- Text and graphic displays showing:
  - CPU and DASD utilizations,
  - Main storage required
  - Number of scanners and Image printers needed
  - Token ring LAN line utilization
  - Optical batch throughputs
  - Image display response time
  - Scanning response time
  - Optical subsystem response time
- A workload file that can be input to QuickSizer/400

C.10.2 How to Receive ImageSizer

The tool can be obtained from PCTOOLS by downloading the QIMG400 Package.

- Link to PCTOOLS disk and enter TOOLCAT PCTOOLS
- Enter ″/QIMG400″ to find QIMG400
- Select the Get option to receive the package files on your VM Reader
- Receive the files from your VM Reader and then download the appropriate worksheet (release of WAF) as a binary file to the personal computer.

LOTUS 1-2-3 (V2.01, V2.2 or later) must be used on the personal computer.

C.11 Performance Evaluation Tool Environment (PETE)

PETE is an internal IBM simulator that, under specific agreement, may be made available to IBM representatives to assist in performance evaluation.

This tool is a client/server simulator that can attach up to 24 users to an AS/400 server. Each of the 24 sessions can be running the same application or up to 24 different applications. All of these sessions originate from one personal computer-type device. The client emulation can be varied based upon such things as keystroke rates and various data entry types.

The personal computer must be a Model 50 or higher with 640KB and a 20 megabyte hard drive. The tool is a DOS-based Micro-Channel PS/2 connected to an OS/2 server. This runs on a LAN to the AS/400.

For more information, contact the IBM Solution Validation Lab in Dallas at 800-742-2493.

C.12 Automated Performance Management Tool (APMT)

C.12.1 General Overview of Function

The repeated running of the Performance Monitor and reviewing its output (for example, use of the Performance Tools printed reports or Advisor output) are often neglected because of the time it takes to analyze the data. Since this document stresses identification of trends before a major problem occurs, a set of programs and objects are offered that significantly automate this process. This set of programs and objects are called the Automated Performance Management Tool (APMT).

The APMT programs can be ordered thru the AS4TOOLS under the TOOLCAT.

These programs and objects are contained in library PFRMGMT. All objects are owned by QPGMR. The source statements for the Control Language, RPG, and Data Description Specifications used are in members within the file named SOURCE. A DFU program (OBJECTIVE) is provided to define the objectives stored in file USROBJ. The programs in PFRMGMT compare the objectives in USROBJ to the profile records in file QACPPROF.

APMT Restriction: Based on QACPPROF File

Note that although the V2R3 and V3R1 Performance Tools/400 System Report continues to support the option to generate performance summary data into the QACPPROF file, this data does not contain performance data for PC Support/400 or Client Access/400 Work Station Function (WSF), PC5250, or RUMBA 5250 jobs. Therefore any work done by these interactive sessions will not be included in any response time trend detection recorded by the APMT support.

If the Performance Monitor data has no dependent workstation jobs (for example only 5250 WorkStation Function (WSF) or RUMBA 5250 emulation jobs), APMT processing will abnormally terminate.

Ensure library PFRMGMT is in the appropriate library list and a program named MEASURE does not already exist in another library within the library list. We have set up two ways of initiating the process so that you may choose the one best suited for the customer situation. The first way is to initiate APMT via start up of a subsystem and the second way is to initiate APMT via a command that can be used at any time:

1. Automated Start: Start subsystem PFRMGMT, which contains an autostart job that sets up the Performance Monitor to run daily at a fixed time. The autostart job will analyze performance data when the Performance Monitor ends.
2. Manual Start: Start the Performance Monitor immediately and start a job that will analyze the performance data when the Performance Monitor ends. This manual start process is initiated via the MSTART command.

If the automatic process is chosen, you should consider stopping the PFRMGMT subsystem over weekends, if no ordinary work is being performed on Saturdays and Sundays. Running measurements on days with no load on the system would give lower averages of response time and so on, and this might incorrectly influence your trend analysis.

The APMT job is shipped to run at priority 21 and will consume a significant amount of CPU utilization when doing trend analysis (before starting the Performance Monitor) and doing performance objectives analysis (after the Performance Monitor ends). This job analyzes measurement profile data and prints Performance Tools reports. You may end the Performance Monitor before the default elapsed time with the ENDPFRMON command.

C.12.2 Intructions to Obtain APMT Code

You can get the APMT code by using the following command in a VM system connected to the IBM VNET network.

```
TOOLS SENDTO RCHVMV RCHTOOLS AS4TOOLS GET APMT PACKAGE
```

The instructions to download the Save File to the AS/400 are included in the package and the instructions to install the tool are covered later in this chapter.

The most efficient way to receive APMT (Save File PFRMGMT in Version 2 Release 2 format) is to have an AS/400 MVS/VM Bridge connection to your IBM network. The MVS/VM Bridge requires licensed program Communications Utilities, 5738-CM1. If this connection exists you can send the AS/400 Save File received in the APMT PACKAGE to a user id on your AS/400. You then do Receive Network File into a Save File on your AS/400. From the Save File you perform the Restore Library command.

No diskette is provided with this publication.

C.12.3 Automated Performance Monitoring Process

Figure 118 shows the general flow for either the automated or manual start of the automated performance monitoring process. The text following the figure describes the process shown.

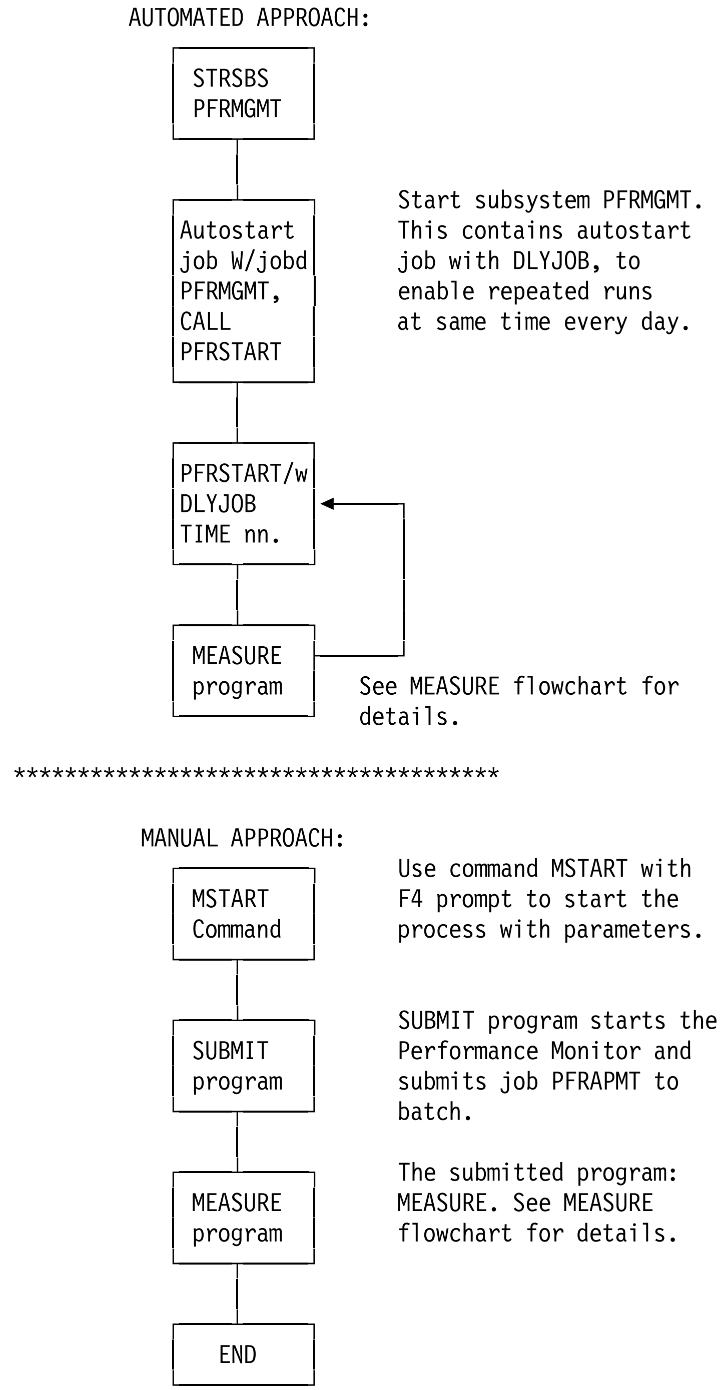


Figure 118. Automated and Manual Start of APMT Process

C.12.4 Automated Performance Monitoring Process Description

The following paragraphs describe the processing performed by the APMT.

1. Check if customer objectives were met last run time, and if **not**, set trace on. (TRACE=*ALL in STRPFRMON command).
2. Check if this is the first time this month that the program is running. If so, perform the trend analysis process (programs TRENDANL and TRENDRPG).

   If a distinct upward trend is detected, messages are sent to QYSYOPR indicating the area, such as response time, where the trend is up. Message APM0002, defined in message file PFRMGMT, is one of the messages sent to QSYSOPR and is defined as alert-capable. The associated Alert Table PFRMGMT and Alert Description are provided should the customer wish to use Alert Support (routing of ″alert messages″) to a remote system for centralized status and assistance.
3. Start the monitor and let it run for the time agreed upon by the customer.
4. Print System report with the option of generating a profile for every measurement. This profile is to be used to evaluate the *trend* of performance on the system.
5. Run the RPG program (OBJTEST) to compare profile data against a file (USROBJ) containing *customer specified objectives*. The key elements being compared are:
   - Interactive response time
   - CPU percent for interactive work
   - Disk utilization percentage.

   If any of these objectives are **not** met, a short ″warning″ report is printed that identifies the component(s) of objectives that were not met, and the number of transactions per hour, together with the customer objective for transactions per hour.
6. If this ″warning report″ is printed, a data area (object type *DTAARA) is updated to make it possible to start the monitor with trace option the next time it is run.
7. The Performance Tools Component and Advisor reports are printed if objectives were **not** met. At the same time a message is sent to the QSYSOPR message queue to inform the operator that the objectives have not been met, and that there are reports to be printed.

   Note that this message (APM0001) is defined in message file PFRMGMT as alert-capable. The associated Alert Table PFRMGMT and Alert Description are provided should the customer wish to use Alert Support (routing of ″alert messages″) to a remote system for centralized status and assistance.

   The alert can be viewed on the focal point system. If that system is an AS/400 the Work with Alert (WRKALR) command can be used to view this alert information.
8. The profile record of today added to the QACPPROF file will be used during the next trend analysis process. If no problem is determined via the automated trend analysis then the performance data members generated by the running of the Performance Monitor are deleted with the Delete Performance Data (DLTPFRDTA) command.

At this time the automated analysis of the profile data does not address all fields that could be analyzed with the profile data. The user of this PFRMGMT support is provided with the source programs to facilitate additions to the automated analysis report. Consider this restriction in the automated process if other data should be analyzed before the detailed performance member data is deleted.

Note that in order to assess communication line utilization and error recovery, either the Performance Tools Print Resource Report (PRTRSCRPT) command or the Advisor output should be reviewed. Prior to Version 2 Release 1 queries were required to analyze communication data collected in the Performance Monitor database files. The Automated Performance Management Tool does not process communications data, as the Performance Tools do not include this data in the measurement profile record information.

9. If the objectives were not met the last time the monitor was run this means this running of the monitor included collecting trace data and therefore a Transaction report is printed automatically, to further increase the possibilities of determining causes of performance problems.

The following pages show the logic flow and program listing for the mainline Automated Performance Management Tool program MEASURE.

Flowchart of CL program: MEASURE

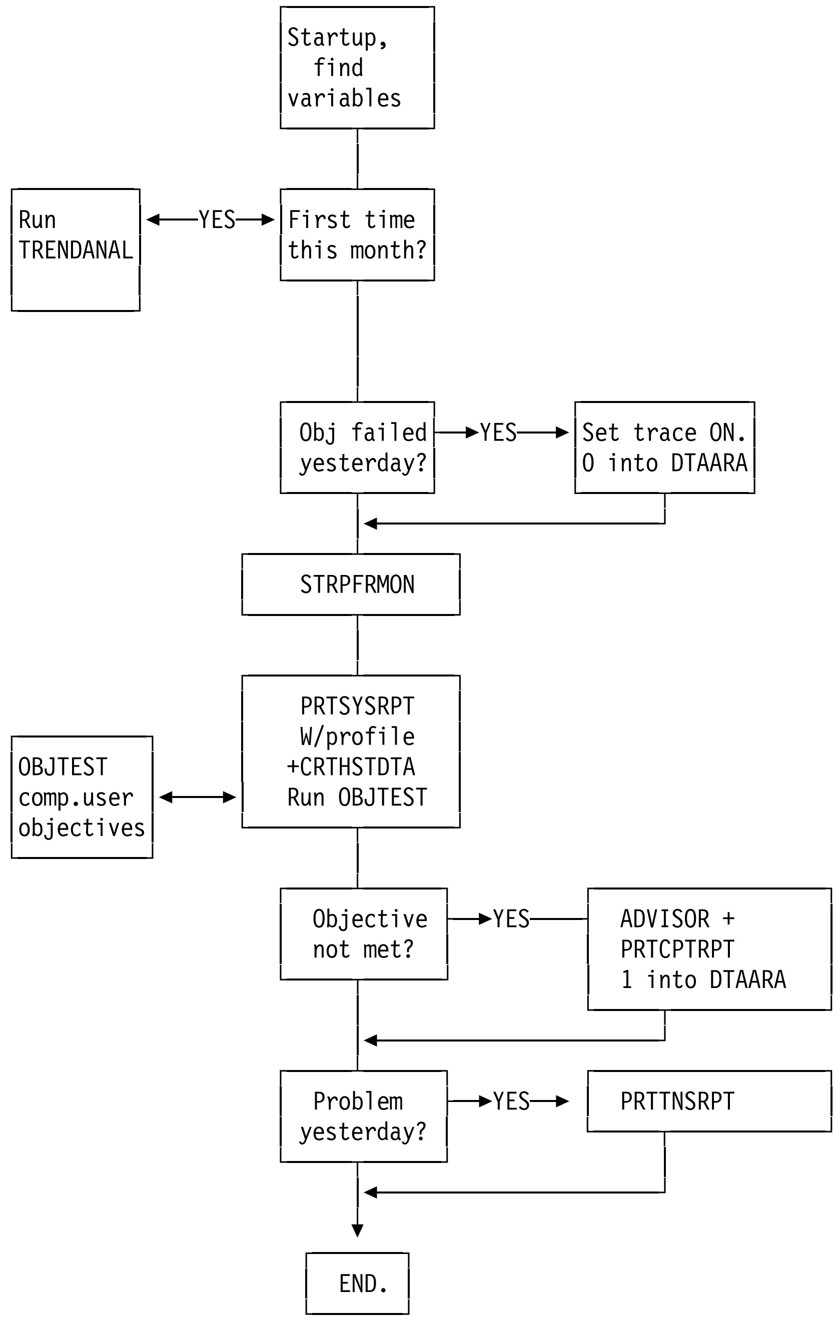


Figure 119. Flowchart of CL Program MEASURE

C.12.5 Listing of Program: MEASURE

```
          /*                                             */
          /*    MAINLINE PROGRAM FOR                     */
          /*      AUTOMATIC PERFORMANCE MANAGEMENT.      */
          /*                                             */
                  PGM        PARM(&INTERVAL &HOUR &MINUTES &TRACE &LIB)
                  DCL        VAR(&HOUR) TYPE(*CHAR) LEN(6)
                  DCL        VAR(&MINUTES) TYPE(*CHAR) LEN(6)
                  DCL        VAR(&INTERVAL) TYPE(*CHAR) LEN(6)
                  DCL        VAR(&TRACE) TYPE(*CHAR) LEN(5)
                  DCL        VAR(&LIB) TYPE(*CHAR) LEN(10)
                  DCL        VAR(&MSGID) TYPE(*CHAR) LEN(10)
                  DCL        VAR(&MSGDTA) TYPE(*CHAR) LEN(100)
                  DCL        VAR(&MBR) TYPE(*CHAR) LEN(10)
                  DCL        VAR(&SWS) TYPE(*CHAR) LEN(8)
                  DCL        VAR(&PROBL) TYPE(*CHAR) LEN(1)
                  DCL        VAR(&THISM) TYPE(*CHAR) LEN(2)
                  DCL        VAR(&PMONTH) TYPE(*CHAR) LEN(2)
                  DCL        VAR(&SYSNAM) TYPE(*CHAR) LEN(10)
          /*  ESTABLISH OUTQ, LOGLEVEL AND RESET SWITCH   */
                  CHGJOB  OUTQ(&LIB/&LIB) LOG(4 00 *NOLIST) SWS(00000000) +
                          RUNPTY(21)
          /*  RETRIEVE CURRENT MONTH                     */
                  RTVSYSVAL  SYSVAL(QMONTH) RTNVAR(&THISM)
          /*                                             */
                  RTVDTAARA  DTAARA(&LIB/MONTH) RTNVAR(&PMONTH)
          /* TREND ANALYSIS BEFORE FIRST RUN EACH MONTH.  */
                  SNDMSG    MSG('TREND ANALYSIS CALLED IF PREVIOUS +
                            MONTH (PFRMGMT/MONTH) DOES NOT EQUAL CURRENT +
                            MONTH (SYSVAL QMONTH). PREV. MONTH:' +
                            *BCAT &PMONTH *CAT '.  CURRENT MONTH:' +
                            *BCAT &THISM *CAT '.')  TOUSR(*SYSOPR)
                  IF         (&PMONTH *NE &THISM) DO
                  CALL       &LIB/TRENDANL PARM(&LIB &PMONTH &THISM)
                  CHGDTAARA  DTAARA(&LIB/MONTH) VALUE(&THISM)
                  ENDDO
          /* START OF NEW MEASUREMENT FOR THIS DAY.     */
                  RTVDTAARA  DTAARA(&LIB/PROBLEM) RTNVAR(&PROBL)
          /* RESET YESTERDAY PROBLEM INDICATION          */
                  CHGDTAARA  DTAARA(&LIB/PROBLEM) VALUE('0')
          /* PROBLEM YESTERDAY = TRACE: *ALL TODAY       */
                  IF         COND(&PROBL *EQ '1')  THEN(DO)
                  CHGVAR     VAR(&TRACE) VALUE('*ALL  ')
                  ENDDO
                  CLRMSGQ    MSGQ(&LIB/PFRMQ)
          /*  MONITOR RUN WITH/WITHOUT TRACE              */
      START:      STRPFRMON  LIB(&LIB) TEXT('PFR.redbook') +
                               INTERVAL(&INTERVAL) HOUR(&HOUR) +
                               MINUTE(&MINUTES) TRACE(&TRACE) +
                               MSGQ(&LIB/PFRMQ)
                  MONMSG     MSGID(CPF0A01) EXEC(DO) /* MONITOR RUNNING ? */
                  DLYJOB     DLY(600)
                  GOTO       START
                  ENDDO
          /*   MONITOR STARTED? GET NEW MEMBER NAME       */
          /*   ENSURE MSGQ(&LIB/PFRMQ) IS CLEAR AT STARTUP */
      NEW:        RCVMSG     MSGQ(&LIB/PFRMQ) WAIT(*MAX) RMV(*YES) +
                               MSGDTA(&MSGDTA) MSGID(&MSGID)
                  IF         COND(&MSGID *EQ 'CPF0A51')  THEN(DO)
```

```
            CHGVAR     VAR(&MBR) VALUE(%SST(&MSGDTA 73 10))
            GOTO NEW
            ENDDO
     /*   MONITOR ENDED ?                          */
            IF         COND(&MSGID *EQ 'CPF0A52')  THEN(GOTO PRINT)
            GOTO NEW
     /*   PRINT SYSTEM REPORT, MAINLY TO GET PROFILE */
 PRINT: PRTSYSRPT  MBR(&MBR) LIB(&LIB) MSRPRF(&MBR) JOBD(*NONE)
     /*   CREATE GRAPHICAL HISTORY DATA FROM PERF. MEMBER */
            CRTHSTDTA  MBR(&MBR) LIB(&LIB) JOBD(*NONE)
            CALL       PGM(OBJTEST) PARM(&MBR)
            RTVJOBA    SWS(&SWS)
     /*   AFTER TEST CUSTOMER OBJECTIVES IN OBJTEST  */
     /*   USER SWITCH (U1) ON IF NOT MET.            */
            IF         (&SWS *EQ '10000000') DO
     /*   IF PROBLEM TODAY: SET PROBL. INDICATION ON */
     /*   AND RUN ADVISOR AND COMPONENT REPORT,      */
     /*   AND SEND MESSAGE TO QSYSOPR OF RESULT.     */
            CHGDTAARA  DTAARA(&LIB/PROBLEM) VALUE('1')
            ANZPFRDTA  MBR(&MBR) LIB(&LIB) OUTPUT(*PRINT)
            QPFR/PRTCPTRPT MBR(&MBR) LIB(&LIB) JOBD(*NONE)
     /* SNDMSG changed to SNDPGMMSG to enable ALERT support, if +
        desired.                                                +
            SNDMSG     MSG('Performance is NOT meeting your +
                         objectives! OBJECTIVE report and ADVISOR +
                         report in outq. PFRMGMT/PFRMGMT.  PLEASE +
                         PRINT.')  TOUSR(*SYSOPR)   */

             RTVNETA SYSNAME(&SYSNAM) /* Place System Name in    +
                       QSYSOPR message. Message APM0001 is alert +
                       capable and the system name assists the +
                       receiver of the associated alert identify +
                       where the report information exists.  */

             SNDPGMMSG MSGID(APM0001) MSGF(PFRMGMT/PFRMGMT)  +
                         MSGDTA(&SYSNAM) TOMSGQ(*SYSOPR) /* Send +
                         the message: "APMT has determined the +
                         performance objectives are not being    +
                         satisfied. Review the Objectives Report +
                         Performance Tools reports on the system." +
                         */
            ENDDO
     /*   IF NO PROBLEM DELETE TODAYS MEMBER BUT KEEP PROFILE */
            IF         (&SWS *EQ '00000000') DO
            QPFR/DLTPFRDTA MBR(&MBR) LIB(&LIB)
            ENDDO
     /*   IF PROBL. YESTERDAY, RUN TRANSACTION REPORT TODAY.  */
            IF         COND(&PROBL *EQ '1')  THEN(DO)
            PRTTNSRPT  MBR(&MBR) TITLE(&MBR) LIB(&LIB) JOBD(*NONE)
            ENDDO
       ENDPGM
```

C.12.6 Installation of PFRMGMT Library

The installation of the library containing the programs and other objects is done be restoring library PFRMGMT. All objects on the diskette are owned by QPGMR and the source for the programs is in file SOURCE.

As shipped, the file QACPPROF contains many records (measurement profiles). This is ideal for verifying installation of APMT, by actually running the tool and having data to be analyzed. However, once deciding to use the APMT, it is recommended that the Clear Physical File Member (CLRPFM) command be used before putting APMT into production mode.

C.12.7 Changing Supplied User Objectives

The user objectives used by this APMT support can be set or changed through a DFU program called OBJECTIVE. The various objectives supported are response time, interactive CPU percent utilization and disk usage utilization. These values are compared to the corresponding values in the profile records for each run.

C.12.8 Changing Supplied Program Parameters

The customer may want to change the defaults used for running the Performance Monitor. These values include run time, interval time and so on. You can change these parameters for the manual start approach by using function key 4 (prompt) on the MSTART command:

	Variable	Default
1. Measurement interval time in minutes	&INTERVAL	10
2. Measurement duration in hours	&HOUR	2
3. Measurement duration in minutes	&MINUTES	0
4. Trace option (*NONE or *all)	&TRACE	*NONE
5. Name of the library that is used.	&LIB	PFRMGMT

You should not change library PFRMGMT unless you wish to make several changes to the programs and messages issued by the PFRMGMT support provided with this publication. Changes would include creating an output queue named with the new library name, creating message queue PFRMQ in the new library, and changing the spooled output information in messages APM0001 and APM0002.

You can also change the parameter values used for the ″manual start approach″ by changing the source description of the MSTART command and recreating the MSTART command. Note that the command processing program for MSTART must be SUBMIT.

To change these same parameters when using the automatic start approach (subsystem PFRMGMT), you need to change the Request Data on the job description PFRMGMT as shown below:

```
CHGJOBD PFRMGMT/PFRMGMT
   RQSDTA('CALL PFRSTART PARM(''10'' ''2'' ''0'' ''*NONE'' ''PFRMGMT'')')
```

The time of day when the automated process starts is in the program PFRSTART. The start time set is at 09:00:00 each day. If you want to start at another time, you will have to change the source of this program, and set the time parameter in the DLYJOB RSMJOB(nnnnnn) command, where nnnnnn is the desired time of day for start of the monitor.

C.12.9 Converting Library PFRMGMT Performance Data

Each new release of the AS/400 may require conversion of performance data collected on a previous release. This applies to the PFRMGMT performance data, including the measured profile data in file QACPPROF.

Because library PFRMGMT contains two logical files (QACPPROL and QACPPROL2) over file QACPPROF, the ″convert performance data″ option of the Performance Tools licensed program will fail. When performance data within library PFRMGMT must be converted you must delete both logical files and then perform the conversion. After the completion of performance data conversion, you must then recreate the logical files QACPPROL and QACPPROL2. The recommended technique is to use the STRPDM (Start Programming Development Manager) command for members in source file SOURCE in library PFRMGMT. Select option 14, Compile, for members QACPPROL and QACPPROL2.

After these logical files have been successfully recreated, the automated performance analysis can be performed on the new release.

C.12.10 Objects in Library PFRMGMT

- DFU: OBJECTIVE

  In the library you will find a DFU program and display file to use for updating the customer objectives for interactive response time, number of transactions per hour, CPU percent for interactive work and disk utilization percent.

  The specifications of the different fields may be found in the USROBJ source physical file. Use the DSPFFD command to display the fields.

- SUBSYSTEM: PFRMGMT

  The subsystem PFRMGMT contains all necessary specifications to run the monitoring as an automated process.

- CL PROGRAM: MEASURE

  This is the mainline program for the Automated Performance Management Tool functions. This program runs in the automated start PFRMGMT subsystem approach or the manual start MSTART command approach.

  The MEASURE program uses a CHGJOB command to establish the default run priority of 21.

- RPG PROGRAM: OBJTEST

  Reads the profile file, and compares the data to the objectives set by the customer (in the Objectives file).

- COMMAND: MSTART

  Input parameters to start the monitoring ″manually″. This command is processed by the SUBMIT CL program.

- CL PROGRAM: SUBMIT

  Submits the MEASURE program to batch for execution.

- CL PROGRAM: PFRSTART

Used in subsystem automatic run process. Called by job description: PFRMGMT, which is defined in the autostart job in the subsystem.

- JOB QUEUE: PFRMGMT
- OUTPUT QUEUE: PFRMGMT
- MESSAGE QUEUE: PFRMQ

  Used to look for messages for start of/end of monitor run. In the Start Performance Monitor message, we find the new member name for this measurement (variable &MBR in the program).

- DATA AREA: PROBLEM

  Changed to ″1″ when customer objectives not met. Signals that next measurement should run with TRACE = *ALL the next time.

- JOB DESCRIPTION: PFRMGMT

  Request data for this job description controls several Performance Monitor parameters.

- PHYSICAL FILE: USROBJ

  This is the user objectives file. Four fields (objectives) are defined: Response Time, Disk Utilization, CPU Utilization, and Transactions per Hour. These fields can be changed with the provided DFU - OBJECTIVE.

- LOGICAL FILEs: QACPPROL, QAPCPROL2

  These files are extracts of the profile file QACPPROF. They are used as input to the RPG program: OBJTEST to make it possible to find the right profile record by key (CHAIN).

- MESSAGE FILE: PFRMGMT

  This message file contains messages APM0001 and APM0002 which are alert-capabile. APM0001 is sent when performance objectives are not met and APM0002 is sent when an upward trend is detected. Alert support for messages can be controlled via the Change Message Description (CHGMSGD) command.

- ALERT TABLE: PFRMGMT

  This Alert Table contains alert descriptions for APM0001 and APM0002. User-selected Alert Code Points for ″Performance Degradation″ are defined for these alert messages. Alert support can be controlled through the Change Network Attributes (CHGNETA) and the Work with Sphere of Control (WRKSOC) commands.

- Graphical formats for CPU, RSP, DISK, TRANS.

  These are contained in graphical package TREND. These will automatically be used to produce graphics when the trend shows increased values over previous averages.

In addition to these objects the various performance measurement files and the performance profile file will be found here instead of in the QPFRDATA library.

Appendix D. Performance Tools/400 Transaction Boundary Overview

D.1 Workstation Transaction Boundaries

The *AS/400 Performance Tools/400 Guide* contains information on transaction boundaries. This is a summary of this information.

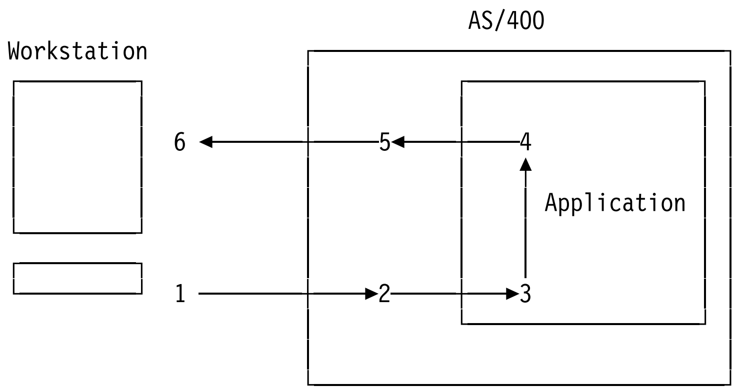


Figure 120. Interactive Transaction Boundary Flow

The numbers in the following discussion refer to Figure 120 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

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, like 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 screen. The following user program operations cause the data to be sent to the screen:

- 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

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 would be included here.

6 System Response 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 would equate 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 would be waiting on:

- an activity level or
- the program to issue an input operation or
- the program to accept input

3 -> 4 Transaction Resources Usage

The period when system resources are used for processing. It would include periods of waiting, like 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 in V2R2 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 to access 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 runtime 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 of files may also benefit but in most cases standard system support for sequential processing or the use of *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 out 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) will be used by your job during run time. You must also understand what other objects will be 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** is issued.

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.

Do not put CLRPOOL and SETOBACC commands for the same storage pool in an initial program run from many workstations. Each issuance of the commands will cause 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 of the storage pool via CLRPOOL could 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 very long time, depending on the number of changed pages that are written. The long run time is most likely 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 will run until the pool has been cleared.

  After the initial usage of CLRPOOL, it is faster to explicitly purge an object using the SETOBJACC command with *PURGE, 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 will be in main storage at the completion of the SETOBJACC command.

  See E.1.2, “SETOBJACC - Determining Object Size” on page 443 for information on object size determination and sample SETOBJACC messages.

- Deleted record space will be loaded into main storage.

  Consider reorganizing the file (RGZPFM command) or the re-use of deleted records (CRTPF command REUSEDLT parameter) before file usage with SETOBJACC.

- Data added to a ″pinned″ file will use 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 would 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.

Then use the Performance Tools/400 Display Access Group (DSPACCGRP) command over the jobs you are interested in. Use the Analyze Access Group (ANZACCGRP) command′s file report to see the 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 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 *MBRLIST parameter to get the member size and then add 30 - 40 KB 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, then use of 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)
  7K of COMPTIME brought to pool with 4000K unused.

> SETOBJACC OBJ(CMN38/CSTFIL) OBJTYPE(*FILE) POOL(CHAINBCH 1)
  2019K 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 4000K bytes (4MB). After COMPTIME was successfully loaded, 3993K remains available for CSTFIL, which is 2019K bytes in in size.

If you then tried to load an object that was larger than 1974K bytes in size, accessing that object may not achieve maximum performance.

E.2 Expert Cache

Expert Cache became available with V2R3, replacing the predecessor V2R2 support - IBM I/O Cache/400 PRPQ (5799-EYG). Expert cache is a selectable option under OS/400 which 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 (CHGSTGPOOL) 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 will be 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 only be used by those who are very knowledgeable on how expert cache works.

Expert cache can be contrasted to the SETOBJACC support where the user must identify what objects should be loaded into main storage and must consider 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 that are physically contiguous within main storage are frequently accessed. 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 are most are likely to see improvements in run time.

In some cases use of 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 pre-determine which files to select and is more cumbersome to use if an application environment changes over time.

Prior to V2R3 the I/O Cache PRPQ (no longer available after V2R2) gave the user similar performance enhancements to Expert Cache but was functionally very different. The PRPQ was **job-oriented** and Expert Cache is **storage pool-oriented**. It also required a detailed analysis of the job′s file processing and experimentation with different parameter values provided under the PRPQ to achieve the desired results.

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 3.6.2, “Memory (Storage) Pools” on page 24

Expert cache uses a storage management tuner, which 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 screen and noting task SMTUNE.

Expert cache dynamically changes the way pages are managed based on pool activity and object usage:

- Files are classified based on 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

- Non-database paging is also affected by the storage management tuner task

  The page fault handler uses tuner information to affect its blocking when page faults occur.

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 (512 bytes), 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 4K up to 128K for database and 32K for non-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 written to disk.
- Call the ″file classer″
- Assess all non-database objects.

Each Data Space and Data Space Index has a different classification for each pool. There are 3 classes defined to the storage management tuner and 1 class defined by database:

- Class 1 - file access appears to be very 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 or 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 will determine how much of that object needs to be brought into the pool (anywhere from 512 bytes to 32 K). Non-DB objects include data queues, data areas, spool files, programs, etc.

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 very 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, then each ODP may have a different access method, but the file will still only have one classification per evaluation period.

Note that in October 1994, the following PTFs were made available that modify the original expert cache algorithms to further improve expert cache performance.

V2R3 PTFs	V3R0.5 PTFs
MF07549, MF07553	MF07566, MF07568

Storage management supports a ″sweeper task″ that ages pages in the storage pool. When started, the task will monitor 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, then the ″sweeper task″ will initiate sufficient aging activity in the pool to reach that rate.

When expert cache is active for a pool, the tuner will set a field in storage management to indicate what blocking factor should be used by the page out tasks to write changed pages. This blocking factor will vary from 4K to 32K based on the blocking that has been done for the read request. The larger the read blocking factor the larger the page out task blocking factor. If the blocking factor changes from 32K to 16K, the page out blocking will stay at 32K for a while to allow the page out task time to get these larger blocks to auxiliary storage. If expert cache is not active for a pool, the blocking factor will remain at the V2R2 default of 4K.

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. IBM Communication Jobs and Subsystem Routing Entries

There are certain environments where customers want to override default work management storage pools and run priorities of IBM software. This section describes how to do this with the *subsystem monitor routing entry compare value support.*

As with all incoming program start requests, the incoming data identifies the initial program to start. With ″customer programs″ these program start requests normally contain character data that can be used with the standard Routing Entry command support. Through the proper compare value specification the incoming program can initiate a job with the desired run priority and within a specified storage pool.

However, most IBM-written program start requests are assigned unique *hexadecimal program names.* Any attempt to code hexadecimal values via the Routing Entry command compare values fails. This is because the underlying system support changes the hexadecimal value received into a character string before invoking the subsystem monitor code that performs routing entry data comparisons.

This section lists the IBM AS/400 function and the appropriate character compare value that can be used to assign a specific storage pool and run priority to the new job. This information can be used when a specific customer environment does not want to use the default storage pool and run priority.

You must use this information carefully or you may create a configuration that degrades, rather than improves overall system performance.

Table 75 (Page 1 of 2). V3R1 IBM Program Routing Entry Compare Values

Compare Value	Compare Start Position	Function
′QOSAPPC ′	37	DIA Version 1
′QZDSTSND ′	37	SNADS SENDER
′QZDRCVR ′	37	SNADS RECEIVER
′QCNTEDDM ′	37	DDM
′QPAPAST2 ′	37	S/36-S/38 PASSTHRU
′QPAPAST2 ′	37	PRINTER PASSTHRU
′QDXPSEND ′	37	DSNX-PC SENDER
′QDXPRCV ′	37	DSNX-PC RECEIVER
′QVPPRINT ′	37	CLIENT ACCESS/400 VIRTUAL PRINT
′QTFDWNLD ′	37	CLIENT ACCESS/400 FILE TRANSFER FACILITY
′QMFRCVR ′	37	CLIENT ACCESS/400 MESSAGE SENDER
′QMFSNDR ′	37	CLIENT ACCESS/400 MESSAGE RECEIVER
′QCNPCSUP ′	37	CLIENT ACCESS/400 SHARED FOLDERS 0, 1
′QTIHNPCS ′	37	ECS TIE FACILITY
′QOQSESRV ′	37	DIA VERSION 2
′QPWFSTP0 ′	37	PC SUPPORT/400 SHARED FOLDERS TYPE 2
′QNMEVK ′	37	SYSTEMS MANAGEMENT UTILITIES
′QHQTRGT ′	37	CLIENT ACCESS/400 REMOTE DATA QUEUE
′QRQSRV1 ′	37	REMOTE SQL - DRDA LEVEL 2 COMMIT

Table 75 (Page 2 of 2). V3R1 IBM Program Routing Entry Compare Values		
Compare Value	**Compare Start Position**	**Function**
′QRQSRV0 ′	37	REMOTE SQL - DRDA LEVEL 2 NO COMMIT
′QND5MAIN ′	37	APPN 5494 APPC CONTROL UNIT
′QCNTEDDM ′	37	REMOTE SQL - DRDA LEVEL
′QRQSRVX ′	37	REMOTE SQL - CONVERGED SERVER
′QOKCSUP ′	37	DIRECTORY SHADOWING
′QS2STSND ′	37	SNADS FS2 SENDER
′QS2RCVR ′	37	SNADS FS2 RECEIVER
′AMQCRC6A ′	37	MESSAGE QUEUING
′QOCEVOKE ′	37	CROSS-SYSTEM CALENDARING
′QLZPSERV ′	37	CLIENT ACCESS/400 LICENSE MANAGER (ORIGINAL CLIENTS)
′QEVYMAIN ′	37	ENVY/400 SERVER
′QACSOTP ′	37	APPC SIGN ON SECURITY
′QZDAINIT ′	37	CLIENT ACCESS/400 DATA ACCESS SERVER
′QNPSERVR ′	37	CLIENT ACCESS/400 NETWORK PRINT SERVER
′QZHQTRG ′	37	CLIENT ACCESS/400 REMOTE DATA QUEUE SERVER
′QZRCSRVR ′	37	CLIENT ACCESS/400 REMOTE COMMAND SERVER
′QZSCSRVR ′	37	CLIENT ACCESS/400 CENTRAL SERVER
′QANRTP ′	37	ADSM/400 APPC CONFIGURATION
′QPWFSTP1 ′	37	CLIENT ACCESS/400 FILE SERVER
′QPWFSTP2 ′	37	CLIENT ACCESS/400 WINDOWS 3.1 FILE SERVER

Note:

- Some compare value table entries are duplicated as they are used for multiple remote transaction requests.
- There are some additional subsystem routing entries that use the APPC mode description (compare value starting position 1) instead of the program name as the compare value. See the index entries for the *work management topic* for various client server applications described in this redbook for additional information.

Appendix G. PC Support/400 Shared Folder Type 2 Performance Query

In V2R2 and V2R3 PC Support/400 shared folder type 2 provides the fastest file serving performance. However, it is difficult to associate the system CPU and disk utilization with the shared folder OS/400 jobs that open and close the shared folder files and LIC tasks (#FSnnnn) that actually exchange data between the AS/400 and the client PC. For those interested in grouping shared folder work for capacity planning purposes, this appendix provides information on how to ″join″ the OS/400 jobs and LIC tasks that work together and sample query output of the appropriate Performance Monitor files.

The information in this topic is completely obsoleted in V3R1.

For V3R1 most LIC CPU and disk resources are assigned to the appropriate OS/400 Client Access/400 job. However, a detailed analysis of V3R1 Client Access/400 OS/400 jobs and associated LIC tasks has not been done in time for publication of this redbook. A specific V3R1 client server performance residency is scheduled for the end of 1995 to determine detail performance management information for the V3R1 Client Access/400 implementation.

Performance information for shared folder type 2 is included in file QAPMTSK and also in the QAPMJOBS file. In the *Work Management Guide* important QAMPTSK file fields are identified as ″internal transaction data ... no field and byte data″. In the QAPMJOBS file, the fields that are relevant to shared folders are ″for IBM internal use only″. This chapter shows how to use these ″internal use only fields″ to determine shared folder system work.

Redbook *PC Support/400 Implementation and Performance*, GG24-3636, provides a description of Type 2 Shared Folders. There, the key system components supporting type 2 shared folders are explained. In this topic we show how to use Performance Monitor data to identify and correlate configuration and job information with the performance data collected.

The key components we will be looking at are:

- The microcode server tasks - FSnnnn
- The XPF Server job - QXFSERV
- The associated station (#Bnnnn) and line I/O manager microcode tasks.
- Related token ring station statistics

G.1 The QAPMTSK File

The entries in the QAPMTSK file provides information that allows correlations to be made to other performance data base files for type 2 shared folders, and also for source and target pass through activity. Field contents will be described for all of these cases; however, further explanation of source and target pass through performance data will not be provided.

In QAPMTSK:

INTNUM Performance Monitor sample interval number. Corresponds to the interval number in other files of the performance data base.

DTETIM Date and time of the sample interval

TYPE Task type

- 01 - source pass through
- 02 - target pass through
- 04 - type 2 shared folders

FLD1 Task name

- #Snnnn - source pass through
- #Tnnnn - target pass through
- FSnnnn - shared folder

Note: For the following fields, 01, 02, 04 is a reference to the contents of the TYPE field described above, e.g. FLD2 coutains the source device name for TYPE = 01 (source passthrough), the target device name for TYPE = 02 (target passthrough), or the server job name for TYPE = 04 (shared folders).

FLD2 Source device (01), target device (02), server job name (04)

FLD3 Source controller (01), target controller (02), job number (04)

FLD4 PC Userid (04)

FLD5 Source LU6.2 device name (01), target LU6.2 device name (02), PC LU6.2 device name (04)

FLD6 Source LU6.2 controller (01), target LU6.2 controller (02), LU6.2 controller for PC (04)

FLD7 Line name

Note: The following fields are only valid for the indicated contents of the TYPE field. The contents of these fields should be ignored when the TYPE field contents are not identified below.

FLD8 Source jobname (01), target jobname (02)

FLD9 Source job number (01), target job number (02)

FLD10 Source user (01), target user (02)

FLD11 Target device (01), source device (02)

FLD12 Target controller (01), source controller (02)

FLD13 Conversation id in hex (04), not displayable

Figure 121 on page 454 shows some sample output created by query FFSERVER, whose definition is provided later in this appendix. The contents of the QAPMTSK file are shown for all active file server tasks during the period of the performance data collection.

Following is an explanation of this data:

- The interval number (INTNUM) is the sampling interval in which the FSnnnn task started. There may not be entries in QAPMTSK for each interval that the server task is active. However, by using the server task names as a criteria for joining with the QAPMJOBS file, it is possible to determine the interval span in which the server task is active (this will be shown later).

- If a file server task starts and terminates in the same interval, its activity will not be identified in the QAPMTSK file.
- The same file server task name may be associated with different jobs in different intervals. For example, task FS0013 is associated with job number 669286 in interval 1 and with job number 670189 in interval 3.
- The COUNT summary function shows the number of active server tasks in each interval. For example, 39 server tasks started in interval 1.
- The server task can be associated with its corresponding QXFSERV job; PC user id; and device, controller, and line descriptor names.

INTNUM	DTETIM	TYPE 04 = File Server	Task Name	Job Name Of Associated Job	Job Number	PC User Id	Device Descriptor Name	Controller Name	Line Descriptor Name
1	931006090122	04	FS000A	QXFSERV	669026	PRINISKI	PTP	PRINISKI	TRNLIN031
	931006090122	04	FS000E	QXFSERV	669212	XZWS766	XZWS766	XZWS766	TRNLIN031
	931006090122	04	FS000F	QXFSERV	669121	KLC	KLC	KLC	TRNLIN031
	931006090122	04	FS0001	QXFSERV	668913	DONNAW	DONNAW	DONNAW	TRNLIN031
	931006090122	04	FS0003	QXFSERV	669069	AJM	AJM	AJM	TRNLIN031
	931006090122	04	FS0004	QXFSERV	669102	DINA	DINA	DINA	TRNLIN031
	931006090122	04	FS0005	QXFSERV	668950	PAULAW	PAULAW	PAULAW	TRNLIN031
	931006090122	04	FS0006	QXFSERV	668998	GILLIS	GILLIS	GILLIS	TRNLIN031
	931006090122	04	FS0008	QXFSERV	669003	LYLEJ	LYLEJ	LYLEJ	TRNLIN031
	931006090122	04	FS001A	QXFSERV	669320	XZWS45F	XZWS45F	XZWS45F	TRNLIN031
	931006090122	04	FS001D	QXFSERV	669358	RIPSTRA	LANJCR	RIPSTRA	TRNLIN031
	931006090122	04	FS001E	QXFSERV	669364	BERGIE	BERGIE	BERGIE	TRNLIN031
	931006090122	04	FS001F	QXFSERV	669384	VIC	VIC	VIC	TRNLIN031
	931006090122	04	FS0010	QXFSERV	669180	NICHOLST	NICHOLST	NICHOLST	TRNLIN031
	931006090122	04	FS0011	QXFSERV	669243	BRENDAP	BRENDAP	BRENDAP	TRNLIN031
	931006090122	04	FS0012	QXFSERV	669269	JPIERCE	JPIERCE	JPIERCE	TRNLIN031
	931006090122	04	FS0013	QXFSERV	669286	WPA	WPA	WPA	TRNLIN031
	931006090122	04	FS0014	QXFSERV	669288	MAYERN	MAYERN	MAYERN	TRNLIN031
	931006090122	04	FS0016	QXFSERV	669291	TRIMBO	TRIMBO	TRIMBO	TRNLIN031
	931006090122	04	FS0018	QXFSERV	669299	LIMPERT	LIMPERTOO	LIMPERT	TRNLIN031
	931006090122	04	FS0019	QXFSERV	669337	REUSCHE	REUSCHE	REUSCHE	TRNLIN031
	931006090122	04	FS002A	QXFSERV	669507	SULLIVAN	MAS	SULLIVAN	TRNLIN031
	931006090122	04	FS002C	QXFSERV	669497	DIANET	DIANET	DIANET	TRNLIN031
	931006090122	04	FS002D	QXFSERV	669499	CAM	CAM	CAM	TRNLIN031
	931006090122	04	FS0020	QXFSERV	669406	WSCHWANE	WSCHWANE	WSCHWANE	TRNLIN031
	931006090122	04	FS0022	QXFSERV	669417	XZWS44A	XZWS44A	XZWS44A	TRNLIN031
	931006090122	04	FS0023	QXFSERV	669434	RAPP	RAPP	RAPP	TRNLIN031
	931006090122	04	FS0024	QXFSERV	669442	DYKSTAL	DYKSTAL	DYKSTAL	TRNLIN031
	931006090122	04	FS0025	QXFSERV	669454	JCHK	JCHK	JCHK	TRNLIN031
	931006090122	04	FS003C	QXFSERV	669691	SCARTER	SCARTER	SCARTER	TRNLIN031
	931006090122	04	FS003E	QXFSERV	669703	XZWS47M	XZWS47M	XZWS47M	TRNLIN031
	931006090122	04	FS003F	QXFSERV	669733	JACOBSEN	BOBJOO	BOBJ	TRNLIN031
	931006090122	04	FS0030	QXFSERV	669515	WEAVER	WEAVER	WEAVER	TRNLIN031
	931006090122	04	FS0032	QXFSERV	669526	MCBRIDE	TMCBRIDE	TMCBRIDE	TRNLIN031
	931006090122	04	FS0034	QXFSERV	669595	JHANSEN	JHANSEN	JHANSEN	TRNLIN031
	931006090122	04	FS0035	QXFSERV	669647	MINNOH	MINNOH	MINNOH	TRNLIN031
	931006090122	04	FS0039	QXFSERV	669656	BAARTMAN	BAARTMAN	BAARTMAN	TRNLIN031
	931006091557	04	FS004A	QXFSERV	669976	JEK	JEK	JEK	TRNLIN031
	931006090122	04	FS0041	QXFSERV	669736	GEBARRY	GEBARRY	GEBARRY	TRNLIN031
	931006090122	04	FS0043	QXFSERV	669742	BEAR	BEAR	BEAR	TRNLIN031
	931006090122	04	FS0044	QXFSERV	669782	STRUBEL	STRUBEL	STRUBEL	TRNLIN031
	931006090122	04	FS0046	QXFSERV	669832	MCKEEHEN	MCKEEHEN	MCKEEHEN	TRNLIN031
	931006090122	04	FS0047	QXFSERV	669840	VALKAL	VALKAL	VALKAL	TRNLIN031
	931006090122	04	FS0048	QXFSERV	669863	WENDYS	WENDYS	WENDYS	TRNLIN031
	931006090122	04	FS0049	QXFSERV	669892	SHER	SHER	SHER	TRNLIN031
		COUNT 45							
2	931006093056	04	FS0047	QXFSERV	670093	VALKAL	VALKAL	VALKAL	TRNLIN031
		COUNT 1							
4	931006100053	04	FS0013	QXFSERV	670189	KMOE	KLMOO	KMOE	TRNLIN031
	931006100053	04	FS004C	QXFSERV	670213	ELLIE	ELLIEOO	ELLIE	TRNLIN031
	931006100053	04	FS004E	QXFSERV	670226	WPA	WPA	WPA	TRNLIN031
	931006100053	04	FS0047	QXFSERV	670231	VALKAL	VALKAL	VALKAL	TRNLIN031
		COUNT 4							
5	931006101551	04	FS004E	QXFSERV	670326	XZWS997	XZWS997	XZWS997	TRNLIN031
		COUNT 1							
6	931006103050	04	FS0008	QXFSERV	670399	MWHALEN	MWHALEN	MWHALEN	TRNLIN031
	931006103050	04	FS0047	QXFSERV	670343	VALKAL	VALKAL	VALKAL	TRNLIN031
		COUNT 2							
7	931006104549	04	FS004A	QXFSERV	670449	JEK	JEK	JEK	TRNLIN031
		COUNT 1							
		FINAL TOTALS COUNT 54							

Figure 121. QAPMTSK Contents For Type 2 Shared Folders

G.2 Correlation Of Shared Folder Data

With the information contained in QAPMTSK, it is possible to establish correlations with performance data in other files of the performance data base. Figure 122 shows the correlations that can be made:

- File server task name in QAPMTSK to the corresponding file server task name in the JBNAME field in QAPMJOBS
- QXFSERV Job name, job number, and PC userid in QAPMTSK to the corresponding names in the JBNAME, JBNBR, and JBUSER fields in QAPMJOBS
- Controller and Line descriptor names in QAPMTSK to the corresponding names in the SLSTNN and SLLND fields in the QAPMSTNL file
- Controller and Line descriptor names in QAPMTSK to the corresponding names in the SCTLNM and SLINNM fields in the QAPMSNA file.

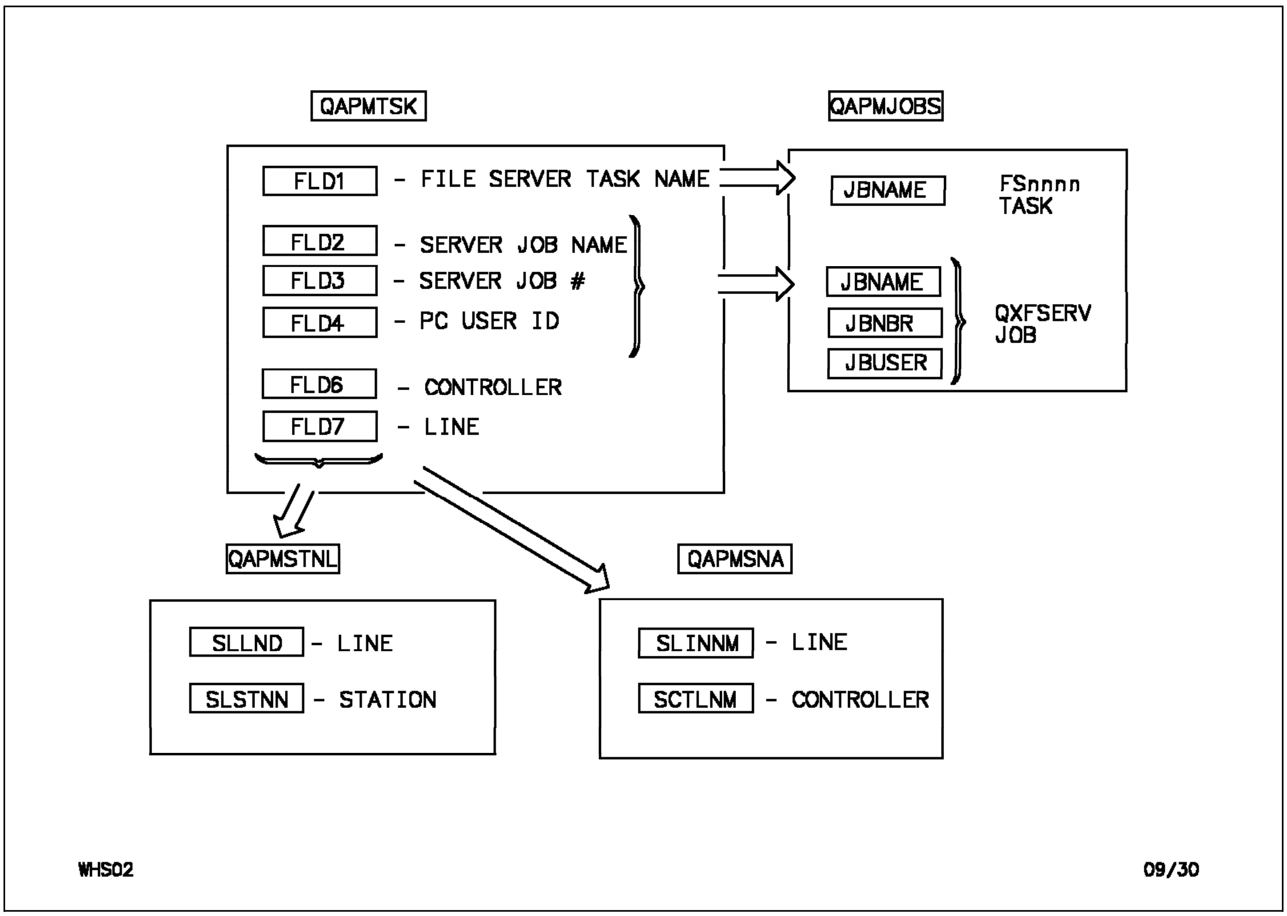


Figure 122. QAPMTSK Correlations With Other Files

G.3 File Server Measurements

File server measurement data is provided for requests that are handled by the file server microcode task. Also, queueing time between the file server task and the SNA station IOM (#Bnnnn) task is measured. Figure 123 on page 457 shows the flow of a request between a PC application and the components on the AS/400 where the measurements are taken. In Figure 123 on page 457 (the numbers in the following list correspond to the numbers in the figure):

1. The PC application sends a request to the AS/400. A PC function, like a ″copy″ will result in multiple requests being handled, and measured, by the file server microcode task. On the AS/400, the request comes through the controller task (SNA station IOM (#Bnnnn)) for this PC.

2. The request is sent to the file server microcode task (FSnnnn). This point in Figure 123 on page 457 identifies the start of the file server task transaction. All requests, or file server task transactions, that are handled entirely by the file server task are (a) counted, and (b) have a cumulative execution time, or response time, kept in the QAPMJOBS record for the file server task (fields where these measurements are kept will be discussed below). ″Cumulative execution time″ is the elapsed time between start and end points of each file server task transaction that is added to a running total for all transactions previously executed.

   File server task transactions that are counted and have a cumulative execution time kept are:

   - Reads and Writes - length of execution is a function of the amount of I/O activity associated with the request.
   - Locks/Unlocks
   - Change end of file, Force Buffers, Resets of files

   Note: Only one bucket is kept that counts and measures the above file server transactions as a group. These ″transactions″ provide a measure of work activity being performed in a file server task that resulted from function that was initiated on a client PCS workstation.

3. End of the file server task transaction.

4. Identifies requests that are handled by the QXFSERV job. These requests are not counted or measured; however, it will be shown below how QXFSERV performance data (e.g. processor usage, I/O activity) can be joined with file server task data to provide a broader picture of file server performance.

   Requests handled by the QXFSERV job are:

   - Create/Delete
   - Open/Close
   - Directory (list file attributes), Make Directory, Remove Directory

5. Identifies a measure of server queueing time between the file server task and the controller task. The file server task sends a reply to the controller task, which is acknowledged by the controller task. The performance data that is kept here is a cumulative queuing time for all sends to the controller task. The following points relate to this measure of queueing time:

   - The measurement starts with the send to the controller and ends with the returned acknowledgement. Elapsed time here includes the link level response from the adjacent system (the PC), and therefore includes controller and line IOM time, communications IOP time, line time, and time in the adjacent system to return the link level response.

- This queueing measurement includes requests that are handled by both the file server task and the QXFSERV job.
- The end of transaction measurement point for file server task transactions typically occurs after the queueing measurement end point.

6. A reply is returned to the PC application.

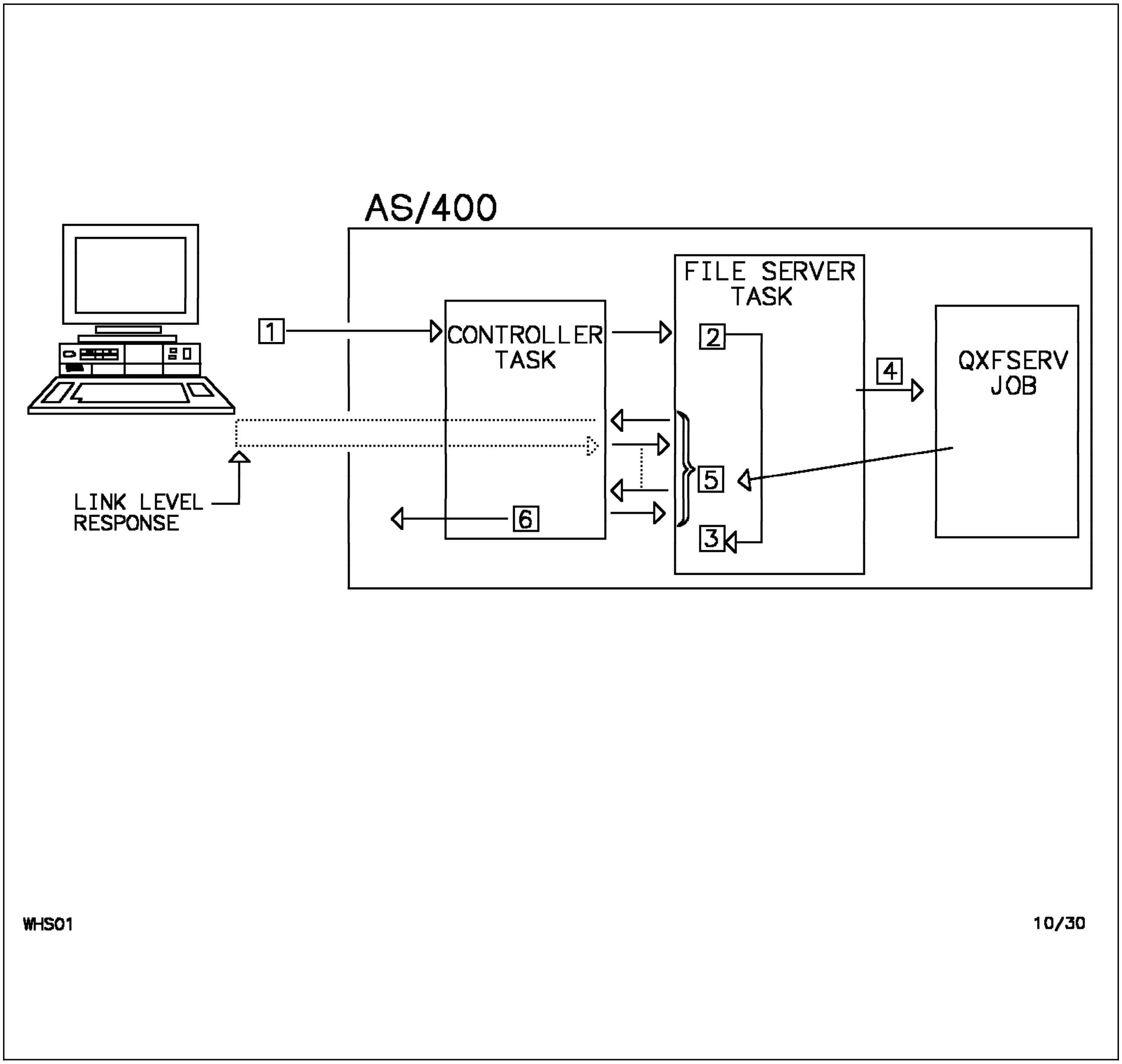


Figure 123. File Server Flow, Measurement Points

G.4 Recycling of FSnnnn Task Names

FSnnnn file server task names are reused by the system when a QXFSERV job and it′s associated file server task terminate. The effect of this reuse of names is evident in the QAPMJOBS file, and should be understood in order to make the correct associations between file server task performance data, the QXFSERV job activity, and the PC user.

In Figure 121 on page 454, note that the file server task name FS004E has entries in both interval 4 and interval 5 in the QAPMTSK file. Remember that QAPMTSK only has an entry for the interval in which the file server task started, and not for every interval that an instance of this task is active. This is, therefore, an indication that the file server task, with which the name FS004E is associated, terminated; the task name FS004E is being reused in the time period represented by intervals 4 and 5.

Further evidence of this is provided in the QAPMTSK contents for the FS004E task - note that the associated QXFSERV job number has changed (from 670226 in interval 4 to 670326 in interval 5); also the PC User Id is different.

Figure 124 on page 459 shows the QAPMJOBS output that corresponds to the entries in QAPMTSK described above for file server task FS004E and the QXFSERV jobs with which FS004E is associated - Job number 670226, and Job number 670326. First, a refresher on the content of the QAPMJOBS ″elapsed interval seconds″ field for jobs and microcode tasks:

- For jobs, it identifies the actual length of time that the job ran in the performance monitor sampling interval.
- For microcode tasks,
  1. In the interval in which the task started, the field only contains the length of the sampling interval, not the amount of time the task ran in that interval.
  2. In the interval in which the task ended, the field identifies the number of seconds into the interval when the task terminated.
  3. If the task started and ended in the same sampling interval, the task actually ran somewhere between 0 seconds and the value shown in the ″elapsed interval seconds″ field.

This output shows that:

- For FS004E entries, the interval 4 and (first) interval 5 entries are associated with QXFSERV job 670226; the (second) interval 5 entry and entries 6-8 are associated with QXFSERV job 670326.
- For the intervals 4,5 for FS004E that are associated with QXFSERV job 670226:
  – It can be assumed that the period of activity for FS004E in interval 4 corresponds to the 348 seconds that the QXFSERV job was active in interval 4.
  – FS004E terminated 786 seconds into interval 5. This value corresponds to the same number for interval 5 for job 670226.
- For the intervals 5-8 for FS004E that are associated with QXFSERV job 670326:

- The period of activity for FS004E in interval 5 corresponds to the 105 seconds that the QXFSERV job was active in interval 5.
- FS004E terminated 599 seconds into interval 8. This value corresponds to the same number in interval 8 for job 670326.

Job Number	Interval Number	Interval Date Time	Elapsed Interval Seconds	Job Name	Job User
	4	931006100053	898	FS004E	
	5	931006101359	786	FS004E	
	5	931006101551	898	FS004E	
	6	931006103050	899	FS004E	
	7	931006104549	898	FS004E	
	8	931006105547	599	FS004E	
670226	4	931006100053	348	QXFSERV	WPA
	5	931006101400	786	QXFSERV	WPA
670326	5	931006101551	105	QXFSERV	XZWS997
	6	931006103050	899	QXFSERV	XZWS997
	7	931006104549	898	QXFSERV	XZWS997
	8	931006105547	599	QXFSERV	XZWS997

Figure 124. Recycling of FSnnnn Task Names - QAPMJOBS Data

G.5 Shared Folder Data in the QAPMJOBS File

The file server task data described above is kept in fields in the QAPMJOBS file in the records for the FSnnnn tasks. The fields containing this data are:

- JBNTR - labeled ″interactive transactions″ in the externally described data definition and identified in the Work Management Guide as having ″a value other than zero only if this is an interactive job or a pass-through target job″. This field, for QAPMJOBS records for FSnnnn tasks, contains the count of file server task transactions as described in item 2 above.
- JBRSP - contains the ″cumulative execution time″, in seconds, for all file server transactions counted in the JBNTR field and as described in item 2 above. Again, this applies to records in the QAPMJOBS file for FSnnnn file server tasks.
- JBQT2 - contains the cumulative queueing time for all file server task and QXFSERV job requests, as described in item 5 above. Note that the value in JBQT2 is in hundredths of a second (i.e. divide by 100 to get the number of seconds).

Figure 125 on page 460 and Figure 126 on page 460 show relevant fields in QAPMJOBS for an active file server microcode task, These figures show the performance data being collected and the results that can be derived from the collected data.

The query definitions for these figures are FFSRVTK1 and FFSRVTK2 in G.6, “Shared Folder Queries” on page 461.

Interval Number	Date	Time	Job Name	CPU Seconds	# File Srvr Transactions	CPU per FS transaction	Total Response Seconds	Avg FS Trans Rsp Time (seconds)	Queueing Time (seconds)	Avg Q′ing Time (seconds)
1	931006	0915	FS0034	14.886	2,946	.005	11	.004	234.070	.079
2	931006	0930		.930	474	.002	3	.006	13.450	.028
3	931006	0945		10.006	2,097	.005	5	.002	160.420	.076
4	931006	1000		15.295	3,235	.005	21	.006	238.960	.074
5	931006	1015		19.107	4,337	.004	22	.005	364.850	.084
6	931006	1030		17.708	4,098	.004	66	.016	298.910	.073
7	931006	1045		21.892	5,775	.004	151	.026	464.100	.080
8	931006	1055		.009	0	++++++++++	0	++++++++	.120	++++++++

Figure 125. QAPMJOBS Contents For Type 2 Shared Folders - Part 1

The query output in Figure 125 shows:

- The intervals in which the file server task is active, i.e. using the processor and/or executing file server transactions.
- CPU Seconds - the number of seconds of processor usage by the file server task in each sampling interval.
- # File Server Transactions - the number of transactions, or requests, (reads, writes, locks, unlocks, etc) executed by the file server task. This is the contents of the JBNTR field.
- CPU per FS transaction - the processor usage per transaction.
- Total Response Seconds - cumulative execution time, or elapsed time, for all file server transactions executed by this file server task. This is the contents of the JBRSP field.
- Avg FS Transaction Response Time - the average transaction execution time, in seconds.
- Queueing Time - total queueing time, in seconds, between the file server task and it′s controller task (SNA Station IOM). This is the contents of the JBQT2 field.
- Average queueing time - this field is derived by dividing the total queueing time by the number of file server transactions. However, since this queueing time also includes activity from the QXFSERV job (whose transactions are <u>not</u> counted), this average should be viewed as an upper bound on the queueing time. Note, again, that this time includes the network turnaround time, including the link level response from the adjacent (PC) system.

Interval Number	Date	Time	Job Name	CPU Seconds	# File Srvr Transactions	Total Reads (non-db, sync+async)	Non-db Reads Per Trans (sync+async)	Total Writes (non-db, sync+async)	Non-db Writes Per Trans (sync+async)
1	931006	0915	FS0034	14.886	2,946	230	.1	497	.2
2	931006	0930		.930	474	6	.0	137	.3
3	931006	0945		10.006	2,097	0	.0	352	.2
4	931006	1000		15.295	3,235	142	.0	958	.3
5	931006	1015		19.107	4,337	103	.0	1,010	.2
6	931006	1030		17.708	4,098	244	.1	2,315	.6
7	931006	1045		21.892	5,775	69	.0	5,471	1.0
8	931006	1055		.009	0	0	+++++++++	0	+++++++++

Figure 126. QAPMJOBS Contents For Type 2 Shared Folders - Part 2

The query output in Figure 126 additionally shows:

- Total synchronous and asynchronous read and write activity in each interval for non-data base I/Os

- Read and write activity per file server transaction, for synchronous and asynchronous non-data base I/Os

G.5.1 File Server Task and QXFSERV Job Relationships

Figure 127 shows the combined resource usage for a file server task and its associated QXFSERV job. This query output is the result of using the QAPMTSK file to establish the relationship between a file server task and its associated QXFSERV job. This query allows the resource usage (processor, total non-data base read/write activity) to be compared. Note in this example that the QXFSERV job is significantly more active than its associated file server task, suggesting the need for more in-depth analysis of the function being performed by the PCS application.

The query that produced this is FILESRVR in G.6, “Shared Folder Queries”

Interval Number	FS Task Name	Job Name	Job Number	FS Task CPU Seconds	QXFSERV CPU Seconds	Total FS Task Reads (non-db)	Total QXFSERV Reads (non-db)	Total FS Task Writes (non-db)	Total QXFSRV Writes (non-db)
1	FS0034	QXFSERV	669595	14.886	60.308	230	455	497	5,400
2		QXFSERV	669595	.930	1.073	6	18	137	237
3		QXFSERV	669595	10.006	39.459	0	2	352	3,522
4		QXFSERV	669595	15.295	53.756	142	463	958	4,941
5		QXFSERV	669595	19.107	65.178	103	205	1,010	5,433
6		QXFSERV	669595	17.708	59.226	244	237	2,315	5,438
7		QXFSERV	669595	21.892	46.095	69	106	5,471	4,102
8		QXFSERV	669595	.009	.089	0	6	0	12

Figure 127. Combined File Server Task, QXFSERV Job Resource Usage

G.6 Shared Folder Queries

This topic provides the query definitions for the type 2 shared folders performance analysis described earlier in this topic. Additional queries are shown that associate type 2 shared folder performance data with SNA (file QAPMSNA) and station (file QAPMSTNL) performance data. All data is collected by the OS/400 Performance Monitor.

G.6.1 FFSERVER Query (File Server Associations)

The FFSERVER query shows the contents of the QAPMTSK file for Type 2 Shared Folders.

- Selects shared folder records (type = ′04′)
- Renames column headings with meaningful names

```
5738QU1  V2R3M0  931217           IBM Query/400           RCHAS1   10/19/93   16:26:47        Page    1


  Query . . . . . . . . . . . . . . . . . FFSERVER
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . File Server Associations - QAPMTSK
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . .
  Query country id  . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . No


Selected files

  ID     File           Library       Member        Record Format

  T01    QAPMTSK        WSRCHAS12     Q932790900    QAPMTSKR


Select record tests

  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')

         TYPE             EQ       '04'


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  INTNUM          10        A           1      INTERVAL NUMBER
  DTETIM                                       INTERVAL DATE AND TIME
  TYPE                                         RECORD TYPE
  FLD1            20        A                  IBM INTERNAL USE
  FLD2                                         IBM INTERNAL USE
  FLD3                                         IBM INTERNAL USE
  FLD4                                         IBM INTERNAL USE
  FLD5                                         IBM INTERNAL USE
  FLD6                                         IBM INTERNAL USE
  FLD7                                         IBM INTERNAL USE
```

Figure 128. FFSERVER Query Definition, Part 1

```
                           IBM Query/400                   10/19/93   16:26:47        Page    2


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        INTNUM                    5    0
  DTETIM                     2        DTETIM                   12
  TYPE                       2        TYPE 04 =                 2
                                      File Server
  FLD1            5          2        Task                     10
                                      Name
  FLD2                       2           Job Name              10
                                      Of  Associated
                                            Job
  FLD3                       2        Job Number               10
  FLD4                       2           PC                    10
                                      User Id
  FLD5                       2          Device                 10
                                      Descriptor
                                         Name
  FLD6                       2        Controller               10
                                         Name
  FLD7                       2           Line                  10
                                      Descriptor
                                         Name


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  1      No    No


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . Yes
    Wrapping width  . . . . . . . . . . .
    Record on one page  . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . WSCHWANES5
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6 (default)
  Report end line . . . . . . . . . . . .  60 (default)
  Report line spacing . . . . . . . . . . Single 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
      File Server Tasks/Job Correlation


Page headings and footings

  Print standard page heading . . . . . . No

    Page heading

    Page footing

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 129. FFSERVER Query Definition, Part 2

G.6.2 FFSRVTK1 Query (FSnnnn Tasks - CPU, Response Time)

Query FFSRVTK1 shows shared folder information from the QAPMJOBS file.

- Defines result fields for cpu per file server transaction, average file server transaction response time, and average queueing time for all FS task and QXFSERV job activity
- Selects QAPMJOBS records for a single active file server task (in this case, for the file server task with name FS0034).

```
5738QU1  V2R3M0  931217           IBM Query/400           RCHAS1   10/19/93   16:35:22        Page    1


  Query . . . . . . . . . . . . . . . . . FFSRVTK1
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . FSnnnn tasks - cpu, rsptime -QAPMJOBS            I
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . .
  Query country id  . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . No


Selected files

  ID     File           Library       Member        Record Format

  T01    QAPMJOBS       WSRCHAS12     Q932790900    QAPMJOBR


Result fields

  Name        Expression                         Column Heading         Len  Dec

  JBCPUSEC    jbcpu/1000                         CPU                      7    3
                                                 Seconds
  CPUPERTRN   jbcpusec/jbntr                     CPU per FS               7    3
                                                 transaction
  IOSUM       jbdbr+jbndb+jbwrt                  TotI/O                   7    0
  TIME        substr(dtetim,7,4)                 Time
  DATE        substr(dtetim,1,6)                 Date
  JBQTIME     jbqt2/100                          Queueing Time           10    3
                                                   (seconds)
  JBFSNTR     jbntr                              # File Srvr             11    0
                                                 Transactions
  AVGQTIME    jbqtime/jbfsntr                    Avg Q'ing                6    3
                                                 Time
                                                 (seconds)
  AVGRSPTIME  jbrsp/jbfsntr                      Avg FS Trans             6    3
                                                 Rsp Time
                                                 (seconds)
  RDSPERTR    jbndb/jbfsntr                      PhysReads                6    0
                                                 Per Trans
                                                  IBM Internal Use Only RCHAS1
  WRTSPERTR   jbwrt/jbfsntr                      PhysWrites               6    0
                                                 Per Trans
```

Figure 130. FFSRVTK1 Query Definition, Part 1

```
                                  IBM Query/400                     10/19/93   16:35:22        Page    2


Select record tests

  AND/OR   Field             Test     Value (Field, Numbers, or 'Characters')

           JBNAME            EQ       'FS0034'
   AND     JBTYPE            EQ       'V'


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  INTNUM          20        A                  Interval Number
  DATE
  TIME
  JBNAME          10        A           1      Job Name
  JBCPUSEC
  JBFSNTR
  CPUPERTRN
  JBRSP                                        Total Response Seconds
  AVGRSPTIME
  JBQTIME
  AVGQTIME


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        Interval                 5    0
                                      Number
  DATE                       2        Date                     6
  TIME                       2        Time                     4
  JBNAME                     1        Job Name                10
  JBCPUSEC                   1        CPU                      7    3
                                      Seconds
  JBFSNTR                    2        # File Srvr             11    0
                                      Transactions
  CPUPERTRN                  2        CPU per FS               7    3
                                      transaction
  JBRSP                      2        Total                   11    0
                                      Response
                                      Seconds
  AVGRSPTIME                 2        Avg FS Trans             6    3
                                      Rsp Time
                                      (seconds)
  JBQTIME                    2        Queueing Time           10    3
                                        (seconds)
  AVGQTIME                   2        Avg Q'ing                6    3
                                      Time
                                      (seconds)


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  0      No    No
  1      No    No
```

Figure 131. FFSRVTK1 Query Definition, Part 2

```
                                   IBM Query/400                    10/19/93   16:35:22         Page    3


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . WSCHWANES5
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6
  Report end line . . . . . . . . . . . .  60
  Report line spacing . . . . . . . . . . Single 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
      VMC File Server Task Activity - QAPMJOBS


Page headings and footings

  Print standard page heading . . . . . . Yes

    Page heading

    Page footing


Database file output

  File  . . . . . . . . . . . . . . . . . XTASKSUMRY
    Library . . . . . . . . . . . . . . . WSGPAC1
  Member  . . . . . . . . . . . . . . . . FRI0929PM
  Data in file  . . . . . . . . . . . . . Replace file
  For a new file:
    Authority . . . . . . . . . . . . . . *CHANGE
    Text about
      the file  . . . . . . . . . . . . . chksum task summary data
  Print definition  . . . . . . . . . . . No


Output file record format

  Output record length  . . . . . . . . .    76

  Field list:

    Field          Begin   Len  Dec   Null  Data Type           Text

    INTNUM             1     5    0         Packed decimal      Interval Number
    DATE               4     6              Character           substr(dtetim,1,6)
    TIME              10     4              Character           substr(dtetim,7,4)
    JBNAME            14    10              Character           Job Name
    JBCPUSEC          24     7    3         Zoned decimal       jbcpu/1000
    JBFSNTR           31    11    0         Zoned decimal       jbntr
    CPUPERTRN         42     7    3         Zoned decimal       jbcpusec/jbntr
    JBRSP             49    11    0         Packed decimal      Total Response Seconds
    AVGRSPTIME        55     6    3         Zoned decimal       jbrsp/jbfsntr
    JBQTIME           61    10    3         Zoned decimal       jbqt2/100
    AVGQTIME          71     6    3         Zoned decimal       jbqtime/jbfsntr


                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 132. FFSRVTK1 Query Definition, Part 3

G.6.3 FFSRVTK2 Query (FSnnnn Tasks - Read/Write per Trans.)

Query FFSRVTK2 shows additional shared folder information from the QAPMJOBS file.

- Defines result fields for total read and write activity for a single active file server task, number of reads and writes per file server task transaction.
- Selects QAPMJOBS records for the same active file server task as was selected in query FFSRVTK1.

```
5738QU1  V2R3M0  931217           IBM Query/400            RCHAS1   10/19/93   16:35:29        Page    1


  Query . . . . . . . . . . . . . . . . . FFSRVTK2
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . FSnnnn tasks, rd/wrt per trans - QAPMJOBS        I
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . .
  Query country id  . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . Hexadecimal


  Processing options

    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . No


Selected files

  ID     File           Library       Member        Record Format

  T01    QAPMJOBS       WSRCHAS12     Q932790900    QAPMJOBR


Result fields

  Name        Expression                        Column Heading         Len  Dec

  JBCPUSEC    jbcpu/1000                        CPU                      7    3
                                                Seconds
  CPUPERTRN   jbcpusec/jbntr                    CPU per FS               7    3
                                                transaction
  IOSUM       jbdbr+jbndb+jbwrt                 TotI/O                   7    0
  TIME        substr(dtetim,7,4)                Time
  DATE        substr(dtetim,1,6)                Date
  JBQTIME     jbqt2/100                         Queueing Time           10    3
                                                  (seconds)
  JBFSNTR     jbntr                             # File Srvr             11    0
                                                Transactions
  AVGQTIME    jbqtime/jbfsntr                   Avg Q'ing                6    3
                                                Time
                                                (seconds)
  AVGRSPTIME  jbrsp/jbfsntr                     Avg FS Trans             6    3
                                                Rsp Time
                                                (seconds)
  RDSPERTR    (jbndb+jbandr)/jbfsntr            Non-db Reads             8    2
                                                Per Trans
                                                (sync+async)
  TOTREADS    jbndb+jbandr                      Total Reads              8    0
                                                (non-db,
                                                sync+async)
  WRTSPERTR   (jbndw+jbandw)/jbfsntr            Non-db Writes            8    2
                                                Per Trans
                                                (sync+async)
  TOTWRITES   jbndw+jbandw                      Total Writes             8    0
                                                (non-db,
                                                sync+async)
```

Figure 133. FFSRVTK2 Query Definition, Part 1

```
                                   IBM Query/400                     10/19/93   16:35:29         Page    2


Select record tests

  AND/OR   Field              Test     Value (Field, Numbers, or 'Characters')

           JBNAME             EQ       'FS0034'
   AND     JBTYPE             EQ       'V'


Ordering of selected fields

  Field            Sort      Ascending/  Break  Field
  Name             Priority  Descending  Level  Text

  INTNUM           20        A                  Interval Number
  DATE
  TIME
  JBNAME           10        A           1      Job Name
  JBCPUSEC
  JBFSNTR
  TOTREADS
  RDSPERTR
  TOTWRITES
  WRTSPERTR


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        Interval                 5    0
                                       Number
  DATE                        2        Date                     6
  TIME                        2        Time                     4
  JBNAME                      1        Job Name                10
  JBCPUSEC                    1        CPU                      7    3
                                       Seconds
  JBFSNTR                     2        # File Srvr             11    0
                                       Transactions
  TOTREADS                    2        Total Reads              8    0
                                       (non-db,
                                       sync+async)
  RDSPERTR                    2        Non-db Reads             8    2          6    1
                                       Per Trans
                                       (sync+async)
  TOTWRITES                   2        Total Writes             8    0
                                       (non-db,
                                       sync+async)
  WRTSPERTR                   2        Non-db Writes            8    2          6    1
                                       Per Trans
                                       (sync+async)


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  0      No    No
  1      No    No
```

Figure 134. FFSRVTK2 Query Definition, Part 2

```
                                   IBM Query/400                      10/19/93   16:35:29        Page    3


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . WSCHWANES5
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6
  Report end line . . . . . . . . . . . .  60
  Report line spacing . . . . . . . . . . Single 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
      VMC File Server Task Activity - QAPMJOBS


Page headings and footings

  Print standard page heading . . . . . . Yes

    Page heading

    Page footing


Database file output

  File  . . . . . . . . . . . . . . . . . XTASKSUMRY
    Library . . . . . . . . . . . . . . . WSGPAC1
  Member  . . . . . . . . . . . . . . . . FRI0929PM
  Data in file  . . . . . . . . . . . . . Replace file
  For a new file:
    Authority . . . . . . . . . . . . . . *CHANGE
    Text about
      the file  . . . . . . . . . . . . . chksum task summary data
  Print definition  . . . . . . . . . . . No


Output file record format

  Output record length  . . . . . . . . .    73

  Field list:

    Field          Begin   Len  Dec   Null  Data Type           Text

    INTNUM             1     5    0         Packed decimal      Interval Number
    DATE               4     6              Character           substr(dtetim,1,6)
    TIME              10     4              Character           substr(dtetim,7,4)
    JBNAME            14    10              Character           Job Name
    JBCPUSEC          24     7    3         Zoned decimal       jbcpu/1000
    JBFSNTR           31    11    0         Zoned decimal       jbntr
    TOTREADS          42     8    0         Zoned decimal       jbndb+jbandr
    RDSPERTR          50     8    2         Zoned decimal       (jbndb+jbandr)/jbfsntr
    TOTWRITES         58     8    0         Zoned decimal       jbndw+jbandw
    WRTSPERTR         66     8    2         Zoned decimal       (jbndw+jbandw)/jbfsntr

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 135. FFSRVTK2 Query Definition, Part 3

G.6.4 FILESRVR Query (File Server Information)

Query FILESRVR uses QAPMTSK to join performance information from QAPMJOBS for a file server task and its associated QXFSERV job.

- Records in QAPMTSK and QAPMJOBS are matched by interval number, file server task name, QXFSERV job name/number and userid
- Records are selected for a single file server task, using only file server entries in QAPMTSK.
- Result fields are defined for total read and write activity for the file server task and the QXFSERV job.

```
5738QU1  V2R3M0  931217          IBM Query/400           RCHAS1   10/19/93   16:38:48        Page    1


  Query . . . . . . . . . . . . . . . . . FILESRVR
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . FileServer info - join QAPMTSK, QAPMJOBS
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . .
  Query country id  . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . Hexadecimal


  Processing options

    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . No


Selected files

  ID     File          Library       Member        Record Format

  T01    QAPMTSK       WSRCHAS12     Q932790900    QAPMTSKR
  T02    QAPMJOBS      WSRCHAS12     Q932790900    QAPMJOBR
  T03    QAPMJOBS      WSRCHAS12     *FIRST        QAPMJOBR


Join tests

  Type of join  . . . . . . . . . . . . . Matched records

  Field              Test              Field

  FLD1               EQ                T02.JBNAME
  FLD2               EQ                T03.JBNAME
  FLD3               EQ                T03.JBNBR
  FLD4               EQ                T03.JBUSER
  T02.INTNUM         EQ                T03.INTNUM


Result fields

  Name        Expression                         Column Heading         Len  Dec

  FSTOTREADS  t02.jbndb+t02.jbandr               Total                   11    0
                                                 FS Task
                                                 Reads (non-db)
  QSTOTREADS  t03.jbndb+t03.jbandr               Total                   11    0
                                                 QXFSERV
                                                 Reads (non-db)
  FSTOTWRTS   t02.jbndw+t02.jbandw               Total                   11    0
                                                 FS Task
                                                 Writes (non-db)
  QXTOTWRTS   t03.jbndw+t03.jbandw               Total                   11    0
                                                 QXFSRV
                                                 Writes (non-db)
  FSCPU       t02.jbcpu/1000                     FS Task                  6    3
                                                 CPU
                                                 Seconds
  QXCPU       t03.jbcpu/1000                     QXFSERV                  6    3
                                                 CPU
                                                 Seconds
```

Figure 136. FILESRVR Query Definition, Part 1

```
                               IBM Query/400                     10/19/93   16:38:48        Page    2


Select record tests

  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')

           T01.TYPE         EQ       '04'
   AND     T01.FLD1         EQ       'FS0034'


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  T02.INTNUM      20        A                  Interval Number
  T02.INTSEC                                   Elapsed Interval Seconds
  T01.FLD1        10        A           1      IBM INTERNAL USE
  T03.JBNAME                                   Job Name
  T03.JBNBR                                    Job Number
  FSCPU
  QXCPU
  FSTOTREADS
  QSTOTREADS
  FSTOTWRTS
  QXTOTWRTS


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

  T02.INTNUM                 0                                 5    0
                                      Interval
                                      Number
  T02.INTSEC                 2        Elapsed                  7    0
                                      Interval
                                      Seconds
  T01.FLD1                   1        FS Task                 10              6
                                      Name
  T03.JBNAME                 2                                10
                                      Job
                                      Name
  T03.JBNBR                  2                                 6
                                      Job
                                      Number
  FSCPU                      2        FS Task                  6    3
                                      CPU
                                      Seconds
  QXCPU                      2        QXFSERV                  6    3
                                      CPU
                                      Seconds
  FSTOTREADS                 2        Total                   11    0
                                      FS Task
                                      Reads (non-db)
  QSTOTREADS                 2        Total                   11    0
                                      QXFSERV
                                      Reads (non-db)
  FSTOTWRTS                  2        Total                   11    0
                                      FS Task
                                      Writes (non-db)
  QXTOTWRTS                  2        Total                   11    0
                                      QXFSRV
                                      Writes (non-db)
```

Figure 137. FILESRVR Query Definition, Part 2

```
                                   IBM Query/400                     10/19/93   16:38:48        Page    3


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  1      No    No


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . *PRINT
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6 (default)
  Report end line . . . . . . . . . . . .  60 (default)
  Report line spacing . . . . . . . . . . Single 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


Page headings and footings

  Print standard page heading . . . . . . Yes

    Page heading

    Page footing

            * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 138. FILESRVR Query Definition, Part 3

G.6.5 FFSRVSNA Query (File Server - SNA Data)

Query FFSRVSNA uses the QAPMTSK file to join data from QAPMJOBS and QAPMSNA to show the relationships between file server task activity and SNA session level traffic.

- Records are matched in QAPMTSK, QAPMJOBS, and QAPMSNA based on job name/number and userid, interval number, and the controller name.
- QAPMTSK records for a single shared folder user are selected
- Result fields include average file server queueing time, RU delivery (service time to adjacent system) time, and average RU length.

```
5738QU1  V2R3M0  931217           IBM Query/400           RCHAS1   10/19/93   16:45:32        Page    1


  Query . . . . . . . . . . . . . . . . . FFSRVSNA
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . FileServer join - TSK, JOBS, SNA
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . .
  Query country id  . . . . . . . . . . .
  Collating sequence  . . . . . . . . . . Hexadecimal


  Processing options

    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . No


Selected files

  ID     File          Library       Member        Record Format

  T01    QAPMTSK       WSRCHAS12     Q932790900    QAPMTSKR
  T02    QAPMJOBS      WSRCHAS12     Q932790900    QAPMJOBR
  T03    QAPMJOBS      WSRCHAS12     *FIRST        QAPMJOBR
  T04    QAPMSNA       WSRCHAS12     *FIRST        QAPMSNAR


Join tests

  Type of join  . . . . . . . . . . . . . Matched records

  Field              Test              Field

  FLD1               EQ                T02.JBNAME
  FLD2               EQ                T03.JBNAME
  FLD3               EQ                T03.JBNBR
  FLD4               EQ                T03.JBUSER
  T02.INTNUM         EQ                T03.INTNUM
  T02.INTNUM         EQ                T04.INTNUM
  FLD6               EQ                T04.SCTLNM
```

Figure 139. FFSRVSNA Query Definition, Part 1

```
                         IBM Query/400                     10/19/93   16:45:32         Page    2


Result fields

  Name        Expression                         Column Heading          Len  Dec

  FSTOTREADS  t02.jbndb+t02.jbandr               Total                    11    0
                                                 FS Task
                                                 Reads (non-db)
  QSTOTREADS  t03.jbndb+t03.jbandr               Total                    11    0
                                                 QXFSERV
                                                 Reads (non-db)
  FSTOTWRTS   t02.jbndw+t02.jbandw               Total                    11    0
                                                 FS Task
                                                 Writes (non-db)
  QXTOTWRTS   t03.jbndw+t03.jbandw               Total                    11    0
                                                 QXFSRV
                                                 Writes (non-db)
  FSCPU       t02.jbcpu/1000                     FS Task                   6    3
                                                 CPU
                                                 Seconds
  QXCPU       t03.jbcpu/1000                     QXFSERV                   6    3
                                                 CPU
                                                 Seconds
  FSQTIME     t02.jbqt2/100                      Total                    10    3
                                                 Q'ingTime
                                                 (seconds)
  AVGQTIME    fsqtime/t02.jbntr                  Avg                       6    3
                                                 Q'ingTime
                                                 (seconds)
  EMAVGTIME   (t04.emtrud/t04.emnrud)/1000       AvgRU Delivery            7    3
                                                 Time To Adj.
                                                 System (M)
  EMAVGLGTH   t04.emlrud/t04.emnrud              Avg RU Lgth              10    0
                                                    (M)
  EMRUPSEC    t04.emnrud/t04.intsec              RUs per                  10    0
                                                 second (M)


Select record tests

  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')

           T01.TYPE          EQ       '04'
   AND     T01.FLD1          EQ       'FS0034'


Ordering of selected fields

  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text

  T02.INTNUM      20        A                 Interval Number
  T02.INTSEC                                  Elapsed Interval Seconds
  T01.FLD1        10        A           1     IBM INTERNAL USE
  T03.JBNAME                                  Job Name
  T03.JBNBR                                   Job Number
  FSCPU
  QXCPU
  FSTOTREADS
  QSTOTREADS
  FSTOTWRTS
  QXTOTWRTS
  FSQTIME
  AVGQTIME
  EMAVGTIME
  T04.SCTLNM                                  Controller Description Name
  T04.SLINNM                                  Line Description Name
  T04.EMNRUD                                  M - RUs Delivered to Adjacent System
  EMAVGLGTH
  T04.EMLRUD                                  M - Length of RUs Delivered to Adjacent System
  T04.EMTRUD                                  M - Service time to Deliver RU to Adjacent System
  EMRUPSEC
  T04.EMNRUR                                  M - RUs Received from Adjacent System
  T04.EMLRUR                                  M - Length of RUs Received from Adjacent System
```

Figure 140. FFSRVSNA Query Definition, Part 2

```
                          IBM Query/400                  10/19/93   16:45:32        Page    3


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

  T02.INTNUM                 0                                 5    0
                                      Interval
                                      Number
  T02.INTSEC                 2        Elapsed                  7    0
                                      Interval
                                      Seconds
  T01.FLD1                   1        FS Task                 10              6
                                      Name
  T03.JBNAME                 2                                10
                                      Job
                                      Name
  T03.JBNBR                  2                                 6
                                      Job
                                      Number
  FSCPU                      2        FS Task                  6    3
                                      CPU
                                      Seconds
  QXCPU                      2        QXFSERV                  6    3
                                      CPU
                                      Seconds
  FSTOTREADS                 2        Total                   11    0
                                      FS Task
                                      Reads (non-db)
  QSTOTREADS                 2        Total                   11    0
                                      QXFSERV
                                      Reads (non-db)
  FSTOTWRTS                  2        Total                   11    0
                                      FS Task
                                      Writes (non-db)
  QXTOTWRTS                  2        Total                   11    0
                                      QXFSRV
                                      Writes (non-db)
  FSQTIME                    2        Total                   10    3
                                      Q'ingTime
                                      (seconds)
  AVGQTIME                   2        Avg                      6    3
                                      Q'ingTime
                                      (seconds)
  EMAVGTIME                  2        AvgRU Delivery           7    3
                                      Time To Adj.
                                      System (M)
  T04.SCTLNM                 2        Controller              10
                                      Description
                                      Name
  T04.SLINNM                 2        Line                    10
                                      Description
                                      Name
  T04.EMNRUD                 2        M - RUs                 11    0
                                      Delivered to
                                      Adjacent System
  EMAVGLGTH                  2        Avg RU Lgth             10    0
                                         (M)
  T04.EMLRUD                 2        M - Length of           11    0
                                      RUs Delivered
                                      to Adjacent System
  T04.EMTRUD                 2        M - Service time        11    0
                                      to Deliver RU
                                      to Adjacent System
  EMRUPSEC                   2        RUs per                 10    0
                                      second (M)
  T04.EMNRUR                 2        M - RUs                 11    0
                                      Received from
                                      Adjacent System
  T04.EMLRUR                 2        M - Length of           11    0
                                      RUs Received
                                      from Adjacent System
```

Figure 141. FFSRVSNA Query Definition, Part 3

```
                              IBM Query/400                     10/19/93   16:45:32        Page    4

Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  1      No    No


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output
  Printer device  . . . . . . . . . . . . *PRINT
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6 (default)
  Report end line . . . . . . . . . . . .  60 (default)
  Report line spacing . . . . . . . . . . Single 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


Page headings and footings

  Print standard page heading . . . . . . Yes

    Page heading

    Page footing

                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 142. FFSRVSNA Query Definition, Part 4

G.6.6 FFSTATION Query (Combined File Server Task Activity)

Query FFSTATION joins performance data for a file server task (from QAPMJOBS), SNA session traffic for the file server activity (from QAPMSNA), and corresponding station statistics (from QAPMSTNL).

- Records are matched based on interval number, file server task job name, station name, and controller name.
- Results calculated include I-frames transmitted and received per second, RUs delivered and received per second, file server task transactions per second.

```
5738QU1  V2R3M0  931217           IBM Query/400           RCHAS1   10/19/93   16:46:49        Page    1


  Query . . . . . . . . . . . . . . . . . FFSTATION
    Library . . . . . . . . . . . . . . . WSCHWANE
  Query text  . . . . . . . . . . . . . . combined FS Task activity - STNL,SNA,JOBS,TSK
  Query CCSID . . . . . . . . . . . . . . 65535
  Query language id . . . . . . . . . . . ENU
  Query country id  . . . . . . . . . . . US
  Collating sequence  . . . . . . . . . . Hexadecimal


  Processing options

    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No (default)
    Ignore substitution warnings  . . . . Yes
    Use collating for all compares  . . . Yes


Selected files

  ID     File          Library       Member       Record Format

  T01    QAPMSTNL      WSRCHAS12     *FIRST       QAPMSTLR
  T02    QAPMSNA       WSRCHAS12     *FIRST       QAPMSNAR
  T03    QAPMJOBS      WSRCHAS12     *FIRST       QAPMJOBR
  T04    QAPMTSK       WSRCHAS12     *FIRST       QAPMTSKR


Join tests

  Type of join  . . . . . . . . . . . . . Matched records

  Field              Test             Field

  T01.INTNUM         EQ               T02.INTNUM
  T01.INTNUM         EQ               T03.INTNUM
  FLD1               EQ               JBNAME
  FLD6               EQ               SLSTNN
  FLD6               EQ               SCTLNM


Result fields

  Name        Expression                        Column Heading          Len  Dec

  ITRANSPSEC  slixmt/t01.intsec                 I-frames                 11    0
                                                Xmit/sec
  IRCVPSEC    slircv/t01.intsec                 I-frames                 11    0
                                                Rcvd/sec
  RUSENTPSEC  t02.emnrud/t02.intsec             RUs delivered            11    0
                                                to Adjacent
                                                Sys/sec
  RURCVDPSEC  t02.emnrur/t02.intsec             RUs Received             11    0
                                                from Adjacent
                                                Sys/sec
  TRANSPSEC   t03.jbntr/t03.intsec              FS Task                  11    0
                                                Transactions
                                                Per Second
```

Figure 143. FFSTATION Query Definition, Part 1

```
                                  IBM Query/400                     10/19/93  16:46:49        Page    2


Select record tests

  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')

           SLSTNN           EQ       'JHANSEN'


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  T01.INTNUM      10        A                  Interval Number
  T01.DTETIM                                   Interval Date and Time in form YYMMDDHHMMSS
  T01.INTSEC                                   Elapsed Interval Seconds
  T01.SLLND                                    Line Description
  T01.SLSTNN                                   Station Name
  T03.JBNAME                                   Job Name
  T01.SLIXMT                                   Total I-Frames Transmitted
  T02.EMNRUD                                   M - RUs Delivered to Adjacent System
  T01.SLIRCV                                   Total I-Frames Received
  T02.EMNRUR                                   M - RUs Received from Adjacent System
  T03.JBNTR                                    Interactive Transactions
  ITRANSPSEC
  RUSENTPSEC
  IRCVPSEC
  RURCVDPSEC
  TRANSPSEC


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

  T01.INTNUM                 0        Interval                  5    0
                                      Number


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
  T01.DTETIM                 2        Interval                 12
                                      Date
                                      and Time
  T01.INTSEC                 2        Elapsed                   7    0
                                      Interval
                                      Seconds
  T01.SLLND                  2        Line                     10
                                      Description
  T01.SLSTNN                 2        Station                  10
                                      Name
  T03.JBNAME                 2        FS Task                  10
                                      Name
  T01.SLIXMT                 2        Total                    11    0
                                      I-Frames
                                      Transmitted
  T02.EMNRUD                 2        M - RUs                  11    0
                                      Delivered to
                                      Adjacent System
  T01.SLIRCV                 2        Total                    11    0
                                      I-Frames
                                      Received
  T02.EMNRUR                 2        M - RUs                  11    0
                                      Received from
                                      Adjacent System
  T03.JBNTR                  2        FS Task                  11    0
                                      Transactions
  ITRANSPSEC                 2        I-frames                 11    0
                                      Xmit/sec
  RUSENTPSEC                 2        RUs delivered            11    0
                                      to Adjacent
                                      Sys/sec
  IRCVPSEC                   2        I-frames                 11    0
                                      Rcvd/sec
  RURCVDPSEC                 2        RUs Received             11    0
                                      from Adjacent
                                      Sys/sec
  TRANSPSEC                  2        FS Task                  11    0
                                      Transactions
                                      Per Second
```

Figure 144. FFSTATION Query Definition, Part 2

```
                              IBM Query/400                  10/19/93   16:46:49        Page    3


 Selected output attributes

   Output type . . . . . . . . . . . . . . Printer
   Form of output  . . . . . . . . . . . . Detail
   Line wrapping . . . . . . . . . . . . . No


 Printer Output

   Printer device  . . . . . . . . . . . . *PRINT
   Report size
     Length  . . . . . . . . . . . . . . .  66 (default)
     Width . . . . . . . . . . . . . . . . 132
   Report start line . . . . . . . . . . .   6 (default)
   Report end line . . . . . . . . . . . .  60 (default)
   Report line spacing . . . . . . . . . . Single space
   Print definition  . . . . . . . . . . . No


 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
       Comparison of Frame and RU Traffic, FS Task Transactions


 Page headings and footings

   Print standard page heading . . . . . . Yes

     Page heading

     Page footing


                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 145. FFSTATION Query Definition, Part 3

Appendix H. Sample X.25 Queries for Network Congestion

The Performance Tools/400 Resource report lists many X.25 statistics but does not include data indicating whether the AS/400 cannot send or receive X.25 frames or the network itself is congested.

The Performance Monitor collects frame congestion data that can be queried to determine if congestion is the reason for poor X.25 performance. Two sample queries and query reports are included in this topic to assist in determining if either the AS/400 or the X.25 network itself is congested (too busy) to process X.25 data within reasonable time periods.

Since the AS/400 or the network may be the cause of congestion, AS/400 CPU utilization, disk utilization, IOP utilization, and storage pool paging information is included in the sample queries. You must review the query output to determine if a performance bottleneck is caused by the AS/400 workload or the network.

Note: These queries were developed and tested against V2R2 Performance Monitor data.

H.1 X25_FRCGT1 Query

X25_FRCGT1 query provides:

- Occurrences of local and remote frames congestion
- Percentage of local and remote frames congestion
- IOP utilization
- Occurrences of buffer-not-available messages received
- Average number of Kbytes of free local storage in the IOP

The following are the definitions for the *X25_FRCGT1 query* . For an example of the output generated for this query, see Figure 149 on page 485. Look for a high percentage under the following query report column headings:

- ″Congestion Local Not Ready (%)″

  A value of 10% or greater indicates the AS/400 is responsible for frame congestion impacting performance.

- Congestion Remote Not Ready (%)″

  A value of 10% or greater indicates the network or remote system is responsible for frame congestion impacting performance.

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   13:39:46        Page    1

  Query . . . . . . . . . . . . . . . . . X25_FRCGT1
    Library . . . . . . . . . . . . . . . ITSCID08
  Query text  . . . . . . . . . . . . . . X.25 Frames Congestion Analysis
  Query CCSID . . . . . . . . . . . . . .    37
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No  (default)
    Ignore substitution warnings  . . . . Yes


Selected files

  ID     File           Library       Member        Record Format

  T01    QAPMX25        QPFRDATA      Q932110838    QAPMX25R
  T02    QAPMCIOP       QPFRDATA      Q932110838    QAPMCIOR


Join tests

  Type of join  . . . . . . . . . . . . . Matched records

  Field              Test              Field

  T01.INTNUM         EQ                T02.INTNUM
  T01.XIOPID         EQ                T02.CIIOP


Result fields

  Name        Expression                        Column Heading          Len  Dec

  IOPUTIL     100 - ((CIIDLC * CIIDLT) /        IOP
              (1000000 * T02.INTSEC))           Util
                                                (%)
  DATE        substr(T01.DTETIM, 3, 2) ||'/' || Date
              substr(T01.DTETIM, 5, 2)
  TIME        substr(T01.DTETIM, 7, 2) ||':' || Time
              substr(T01.DTETIM, 9, 2)
  IOPMEM      (CIRAMU / 1024)                    Available
                                                IOP Storage
                                                   (KB)
  TOTLOCFRM   (XHEFFR + XHFRIE + XHIFR + XHRNRR Total Local Frames
              + XHRRFR)
  TOTREMFRM   (XHIFTR + XHIFRT + XHRNRT +       Total Remote Frames
              XHRRFT)
  LOCNOTRDY   (XHRNRT * 100) / TOTLOCFRM        Congestion
                                                Local  Not
                                                Ready  (%)
  REMNOTRDY   (XHRNRR * 100) / TOTREMFRM        Congestion
                                                Remote Not
                                                Ready  (%)
```

Figure 146. X25_FRCGT1 Query Definition, Part 1

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   13:39:46        Page    2


Select record tests

  AND/OR   Field             Test     Value (Field, Numbers, or 'Characters')
           TOTLOCFRM         GT       0
   AND     TOTREMFRM         GT       0


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  T01.XIOPB       10        A           1      IOP Bus Number
  T01.XIOPA       20        A           1      IOP Bus Address
  T01.XLLND       30        A           2      Line Description
  T01.XLLSP                                    Line Speed
  DATE
  TIME
  T01.XHRNRT                                   RNR Frames Transmitted
  LOCNOTRDY
  T01.XHRNRR                                   RNR Frames Received
  REMNOTRDY
  IOPUTIL
  T02.CIBNAR                                   Occurrences of BNA Received
  IOPMEM


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
  T01.XIOPB                  0         IOP                    3    0
                                        Bus
                                      Number
  T01.XIOPA                  2          IOP                   3    0
                                        Bus
                                      Address
  T01.XLLND                  2        Line                   10
                                      Name
  T01.XLLSP       2          2        Line                   11    0          5    0
                                      Speed
                                      (bps)
  DATE                       2        Date                    5
  TIME                       2        Time                    5
  T01.XHRNRT      2 4        2           RNR                 11    0          5    0
                                        Frames
                                      Transmitted
  LOCNOTRDY       2 4        2        Congestion             16    2          4    1
                                      Local  Not
                                      Ready  (%)
  T01.XHRNRR      2 4        2          RNR                  11    0          5    0
                                       Frames
                                      Received
  REMNOTRDY       2 4        2        Congestion             16    2          4    1
                                      Remote Not
                                      Ready  (%)
  IOPUTIL         2 4        2        IOP                    25    2          4    1
                                      Util
                                      (%)
  T02.CIBNAR      2 4        2        Occurrences            11    0          5    0
                                        of BNA
                                       Received
  IOPMEM          2 3        2         Available             13    2          5    0
                                      IOP Storage
                                         (KB)
```

Figure 147. X25_FRCGT1 Query Definition, Part 2

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   13:39:46        Page    3


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  0      No    No         Summary for all X.25 lines
  1      No    No         Summary for IOP &T01.XIOPA &T01.XIOPB
  2      No    No         Summary for line &T01.XLLND


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . *PRINT
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6 (default)
  Report end line . . . . . . . . . . . .  60 (default)
  Report line spacing . . . . . . . . . . Single space
  Print definition  . . . . . . . . . . . No


Printer Spooled Output

  Spool the output  . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Form type . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Copies  . . . . . . . . . . . . . . . .   1
  Hold  . . . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)


Cover Page

  Print cover page  . . . . . . . . . . . Yes
    Cover page title
                     X.25 Query
              Frames Congestion Analysis
              IOP  Memory Considerations


Page headings and footings

  Print standard page heading . . . . . . Yes

    Page heading

    Page footing


                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 148. X25_FRCGT1 Query Definition, Part 3

H.1.1 X25_FRCGT1 Query - Detailed Results

```
                                              X.25 Query
                                       Frames Congestion Analusis
                                       IOP  Memory Considerations

09/13/93  13:39:37                                                                                                  PAGE    1

 IOP     IOP    Line          Line    Date   Time      RNR      Congestion    RNR     Congestion   IOP    Occurrences   Available
 Bus     Bus    Name          Speed                   Frames    Local  Not   Frames   Remote Not   Util     of BNA     IOP Storage
Number  Address               (bps)                 Transmitted Ready  (%)  Received  Ready  (%)   (%)     Received       (KB)

   2       3    LINX25B       4,800   07/30  08:54        13          .2         0          .0     49.4         0           797
                              4,800   07/30  09:09        61          .5         0          .0     24.3         0           797
                              4,800   07/30  09:24         9          .1         0          .0     23.6         0           797
                              4,800   07/30  09:39        15          .2         0          .0     23.5         0           797
                              4,800   07/30  09:54        13          .1         0          .0     24.9         0           797
                              4,800   07/30  10:09        11          .2         0          .0     23.1         0           797
                              4,800   07/30  10:24         0          .0         0          .0     22.1         0           797
                              4,800   07/30  10:39         9          .1         0          .0     22.5         0           797
                              4,800   07/30  10:54         4          .1         0          .0     22.6         0           797
                              4,800   07/30  11:09         8          .1         0          .0     22.1         0           797
                                               .
                                               .
                              4,800   07/30  17:23         8          .1         0          .0     23.0         0           876
                              4,800   07/30  17:38         1          .0         0          .0     22.4         0           876
                              4,800   07/30  17:53       219         1.4         0          .0     25.1         0           876
                              4,800   07/30  18:08        76          .5         0          .0     25.0         0           876
                              4,800   07/30  18:23        24          .1         0          .0     26.8         0           876
                              4,800   07/30  18:38         3          .0         0          .0     22.1         0           876

                         Summary for line LINX25B
                         AVG  4,800                       16          .2         0          .0     22.5         0           823
                         MIN                                                                                                797
                         MAX                             219         1.4         0          .0     49.4         0

                         Summary for IOP 3 2
                         AVG  4,800                       16          .2         0          .0     22.5         0           823
                         MIN                                                                                                797
                         MAX                             219         1.4         0          .0     49.4         0

------------------------------------------------------------------------------------------------------------------------------

09/13/93  13:39:37                                                                                                  PAGE    2

 IOP     IOP    Line          Line    Date   Time      RNR      Congestion    RNR     Congestion   IOP    Occurrences   Available
 Bus     Bus    Name          Speed                   Frames    Local  Not   Frames   Remote Not   Util     of BNA     IOP Storage
Number  Address               (bps)                 Transmitted Ready  (%)  Received  Ready  (%)   (%)     Received       (KB)

   3       5    LINX25A       9,600   07/30  08:54        13          .2         0          .0     40.6         0           842
                              9,600   07/30  09:09        10          .1         0          .0      8.8         0           843
                              9,600   07/30  09:24        27          .3         0          .0     10.1         0           843
                              9,600   07/30  09:39        33          .2         0          .0     12.3         0           843
                              9,600   07/30  09:54        13          .1         0          .0     12.5         0           843
                              9,600   07/30  10:09        17          .1         0          .0     10.7         0           843
                              9,600   07/30  10:24        30          .2         0          .0     12.2         0           843
                              9,600   07/30  10:39         4          .0         0          .0     12.4         0           843
                              9,600   07/30  10:54         6          .1         0          .0     11.7         0           843
                              9,600   07/30  11:09         5          .1         0          .0      9.2         0           843
                                               .
                                               .
                              9,600   07/30  17:23         2          .0         0          .0     12.6         0           842
                              9,600   07/30  17:38        17          .2         0          .0     14.1         0           842
                              9,600   07/30  17:53       167          .8         0          .0     15.0         0           842
                              9,600   07/30  18:08        75          .5         0          .0     14.6         0           842
                              9,600   07/30  18:23         8          .0         0          .0     14.0         0           842
                              9,600   07/30  18:38         5          .0         0          .0     11.9         0           842

                         Summary for line LINX25A
                         AVG  9,600                       14          .1         0          .0     11.7         0           845
                         MIN                                                                                                842
                         MAX                             167          .8         0          .0     40.6         0

                         Summary for IOP 5 3
                         AVG  9,600                       14          .1         0          .0     11.7         0           845
                         MIN                                                                                                842
                         MAX                             167          .8         0          .0     40.6         0

                         Summary for all X.25 lines
                         AVG  7,200                       15          .1         0          .0     17.1         0           834
                         MIN                                                                                                797
                         MAX                             219         1.4         0          .0     49.4         0

* * *  E N D  O F  R E P O R T  * * *
```

Figure 149. Example Results from X25_FRCGT1 Query - Detailed Report

H.2 X25_FRCGT2 Query

This is the second query regarding with X.25 frames congestion. *X25_FRCGT2 query* provides:

- Occurrences of local and remote frame congestion
- Percentage of local and remote frame congestion
- Machine pool size and occurrences of non-database page faults per second
- CPU and disk arm (arm #1) utilization
- Average response time

The following are the definitions for the *X25_FRCGT2 query* . For an example of the output generated for this query, see Figure 153 on page 490. Look for a high percentage under the following query report column headings:

- ″Congestion Local Not Ready (%)″

  A value of 10% or greater indicates the AS/400 is responsible for frame congestion impacting performance.

- Congestion Remote Not Ready (%)″

  A value of 10% or greater indicates the network or remote system is responsible for frame congestion impacting performance.

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   16:58:43        Page    1

  Query . . . . . . . . . . . . . . . . . X25_FRCGT2
    Library . . . . . . . . . . . . . . . ITSCID08
  Query text  . . . . . . . . . . . . . . X25 Frame Congestion Analysis
  Query CCSID . . . . . . . . . . . . . .    37
  Collating sequence  . . . . . . . . . . Hexadecimal

  Processing options
    Use rounding  . . . . . . . . . . . . Yes (default)
    Ignore decimal data errors  . . . . . No  (default)
    Ignore substitution warnings  . . . . Yes

Selected files

  ID     File           Library       Member        Record Format

  T01    QAPMX25        QPFRDATA      Q932110838    QAPMX25R
  T02    QAPMPOOL       QPFRDATA      Q932110838    QAPMPOOR
  T03    QAPMSYS        QPFRDATA      Q932110838    QAPMSYSR
  T04    QAPMDISK       QPFRDATA      Q932110838    QAPMDISR

Join tests

  Type of join  . . . . . . . . . . . . . Matched records

  Field             Test             Field

  T01.INTNUM        EQ               T02.INTNUM
  T01.INTNUM        EQ               T03.INTNUM
  T01.INTNUM        EQ               T04.INTNUM

Result fields

  Name        Expression                        Column Heading          Len  Dec

  TOTLOCFRM   (XHEFFR + XHFRIE + XHIFR + XHRNRR  Total Local Frames
              + XHRRFR)
  TOTREMFRM   (XHIFTR + XHIFRT + XHRNRT +        Total Remote Frames
              XHRRFT)
  DATE        substr(T01.DTETIM, 3, 2) ||'/' ||  Date
              Substr(T01.DTETIM, 5, 2)
  TIME        substr(T01.DTETIM, 7, 2) ||':' ||  Time
              substr(T01.DTETIM, 9, 2)
  LOCNOTRDY   (XHRNRT * 100) / TOTLOCFRM         Congestion
                                                 Local  Not
                                                 Ready  (%)
  REMNOTRDY   (XHRNRR * 100) / TOTREMFRM         Congestion
                                                 Remote Not
                                                 Ready  (%)
  MCHFAULTS   (PONDBF / T02.INTSEC)               Mch Pool
                                                 NDB Faults
                                                  per sec.
  INTCPUUTIL  ((SDCPU + SWCPU + SPCPU + S6CPU +   Inter
              SICPU) / 10) / T03.INTSEC          CPU Util
                                                   (%)
  ARM1BUSY    ((DSSMPL - DSNBSY) / DSSMPL) *     Arm #1
              100                                 Util
                                                  (%)
  TOTTRN      (SDRES2 + SWRES2 + SPRES2 +        Number Transactions
              S6TRNS + SITRNS)
  TOTTRNTIME  (SDRES1 + SWRES1 + SPRES1 +        Total Transaction
              S6TRNT + SITRNT)                   Time
  RESPTIME    (TOTTRNTIME / TOTTRN)              Avg Rsp
                                                  Time
                                                  (sec)
```

Figure 150. X25_FRCGT2 Query Definition, Part 1

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   16:58:43        Page    2


Select record tests

  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           TOTLOCFRM        GT       0
   AND     TOTREMFRM        GT       0
   AND     PONBR            EQ       '01'
   AND     DSARM            EQ       '0001'


Ordering of selected fields

  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text

  T01.XLLND       10        A           1      Line Description
  T01.XLLSP                                    Line Speed
  DATE
  TIME
  T01.XHRNRT                                   RNR Frames Transmitted
  LOCNOTRDY
  T01.XHRNRR                                   RNR Frames Received
  REMNOTRDY
  T02.POSIZ                                    Pool Size in K Bytes
  MCHFAULTS
  INTCPUUTIL
  ARM1BUSY
  RESPTIME


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

  T01.XLLND                  0        Line                    10
                                      Name
  T01.XLLSP       2          2        Line                    11    0          5    0
                                      Speed
                                      (bps)
  DATE                       2        Date                     5
  TIME                       2        Time                     5
  T01.XHRNRT      2 4        2           RNR                  11    0          5    0
                                        Frames
                                      Transmitted
  LOCNOTRDY       2 4        2        Congestion              16    2          4    1
                                      Local  Not
                                      Ready  (%)
  T01.XHRNRR      2 4        2          RNR                   11    0          5    0
                                       Frames
                                      Received
  REMNOTRDY       2 4        2        Congestion              16    2          4    1
                                      Remote Not
                                      Ready  (%)
  T02.POSIZ       2 4        2        Mch Pool                 7    0
                                        Size
                                        (KB)
  MCHFAULTS       2 4        2         Mch Pool               13    2          3    1
                                      NDB Faults
                                       per sec.
  INTCPUUTIL      2 4        2         Inter                  17    2          4    1
                                      CPU Util
                                        (%)
  ARM1BUSY        2 4        2        Arm #1                  17    2          4    1
                                       Util
                                       (%)
  RESPTIME        2 4        2        Avg Rsp                 17    2          4    1
                                       Time
                                       (sec)
```

Figure 151. X25_FRCGT2 Query Definition, Part 2

```
5738QU1  V2R2M0  920925           IBM Query/400           RCHAS149  9/13/93   16:58:43        Page    3


Report breaks

  Break  New   Suppress   Break
  Level  Page  Summaries  Text

  0      No    No         Summary for all X.25 Lines
  1      No    No         Summary for line &T01.XLLND


Selected output attributes

  Output type . . . . . . . . . . . . . . Printer
  Form of output  . . . . . . . . . . . . Detail
  Line wrapping . . . . . . . . . . . . . No


Printer Output

  Printer device  . . . . . . . . . . . . *PRINT
  Report size
    Length  . . . . . . . . . . . . . . .  66 (default)
    Width . . . . . . . . . . . . . . . . 132
  Report start line . . . . . . . . . . .   6 (default)
  Report end line . . . . . . . . . . . .  60 (default)
  Report line spacing . . . . . . . . . . Single space
  Print definition  . . . . . . . . . . . No


Printer Spooled Output

  Spool the output  . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Form type . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)
  Copies  . . . . . . . . . . . . . . . .   1
  Hold  . . . . . . . . . . . . . . . . . (Defaults to value in print file, QPQUPRFIL)


Cover Page

  Print cover page  . . . . . . . . . . . Yes
    Cover page title
                              X.25 Query
                       Frames Congestion Analysis
                     Machine Storage Considerations
         Interactive CPU and Disk Utilization and Response Time


Page headings and footings
  Print standard page heading . . . . . . Yes
    Page heading
    Page footing


                * * * * *   E N D   O F   Q U E R Y   P R I N T   * * * * *
```

Figure 152. X25_FRCGT2 Query Definition, Part 3

H.2.1 X25_FRCGT2 Query - Detailed Results

X.25 Query
Frames Congestion Analysis
Machine Storage Considerations
Interactive CPU and Disk Utilization and Response Time

09/15/93 10:13:52 PAGE 1

Line Name	Line Speed (bps)	Date	Time	RNR Frames Transmitted	Congestion Local Not Ready (%)	RNR Frames Received	Congestion Remote Not Ready (%)	Mch Pool Size (KB)	Mch Pool NDB Faults per sec.	Inter CPU Util (%)	Arm #1 Util (%)	Avg Rsp Time (sec)
LINX25A	9,600	07/30	08:54	13	.2	0	.0	23,500	3.1	49.7	9.7	3.2
	9,600	07/30	09:09	10	.1	0	.0	23,500	2.6	55.0	9.6	2.7
	9,600	07/30	09:24	27	.3	0	.0	23,500	2.2	62.1	9.9	3.4
	9,600	07/30	09:39	33	.2	0	.0	23,500	1.9	61.9	9.2	3.2
	9,600	07/30	09:54	13	.1	0	.0	23,500	1.5	54.1	6.4	2.4
	9,600	07/30	10:09	17	.1	0	.0	23,500	1.8	50.0	7.0	2.6
	9,600	07/30	10:24	30	.2	0	.0	23,500	1.9	68.7	8.9	4.7
	9,600	07/30	10:39	4	.0	0	.0	23,500	2.0	72.1	10.6	5.2
	9,600	07/30	10:54	6	.1	0	.0	23,500	2.3	73.2	12.2	5.0
	9,600	07/30	11:09	5	.1	0	.0	23,500	2.2	72.4	10.4	5.2
			.									
			.									
			.									
	9,600	07/30	17:23	2	.0	0	.0	23,500	1.8	69.0	9.5	3.3
	9,600	07/30	17:38	17	.2	0	.0	23,500	2.1	71.0	8.9	3.9
	9,600	07/30	17:53	167	.8	0	.0	23,500	2.6	62.0	10.5	2.7
	9,600	07/30	18:08	75	.5	0	.0	23,500	1.7	59.5	7.6	5.4
	9,600	07/30	18:23	8	.0	0	.0	23,500	1.5	46.7	5.6	2.6
	9,600	07/30	18:38	5	.0	0	.0	23,500	1.6	43.4	6.7	2.5
Summary for line LINX25A												
AVG	9,600			14	.1	0	.0	23,500	1.9	54.3	8.5	3.2
MAX				167	.8	0	.0	23,500	3.5	73.2	15.0	5.4

09/15/93 10:13:52 PAGE 2

Line Name	Line Speed (bps)	Date	Time	RNR Frames Transmitted	Congestion Local Not Ready (%)	RNR Frames Received	Congestion Remote Not Ready (%)	Mch Pool Size (KB)	Mch Pool NDB Faults per sec.	Inter CPU Util (%)	Arm #1 Util (%)	Avg Rsp Time (sec)
LINX25B	4,800	07/30	08:54	13	.2	0	.0	23,500	3.1	49.7	9.7	3.2
	4,800	07/30	09:09	61	.5	0	.0	23,500	2.6	55.0	9.6	2.7
	4,800	07/30	09:24	9	.1	0	.0	23,500	2.2	62.1	9.9	3.4
	4,800	07/30	09:39	15	.2	0	.0	23,500	1.9	61.9	9.2	3.2
	4,800	07/30	09:54	13	.1	0	.0	23,500	1.5	54.1	6.4	2.4
	4,800	07/30	10:09	11	.2	0	.0	23,500	1.8	50.0	7.0	2.6
	4,800	07/30	10:24	0	.0	0	.0	23,500	1.9	68.7	8.9	4.7
	4,800	07/30	10:39	9	.1	0	.0	23,500	2.0	72.1	10.6	5.2
	4,800	07/30	10:54	4	.1	0	.0	23,500	2.3	73.2	12.2	5.0
	4,800	07/30	11:09	8	.1	0	.0	23,500	2.2	72.4	10.4	5.2
			.									
			.									
			.									
	4,800	07/30	17:23	8	.1	0	.0	23,500	1.8	69.0	9.5	3.3
	4,800	07/30	17:38	1	.0	0	.0	23,500	2.1	71.0	8.9	3.9
	4,800	07/30	17:53	219	1.4	0	.0	23,500	2.6	62.0	10.5	2.7
	4,800	07/30	18:08	76	.5	0	.0	23,500	1.7	59.5	7.6	5.4
	4,800	07/30	18:23	24	.1	0	.0	23,500	1.5	46.7	5.6	2.6
	4,800	07/30	18:38	3	.0	0	.0	23,500	1.6	43.4	6.7	2.5
Summary for line LINX25B												
AVG	4,800			16	.2	0	.0	23,500	1.9	54.7	8.5	3.3
MAX				219	1.4	0	.0	23,500	3.5	73.2	15.0	6.6
Summary for all X.25 Lines												
AVG	7,200			15	.1	0	.0	23,500	1.9	54.5	8.5	3.3
MAX				219	1.4	0	.0	23,500	3.5	73.2	15.0	6.6

* * * E N D O F R E P O R T * * *

Figure 153. Example Results from X25_FRCGT2 Query - Detailed Report

Appendix I. 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
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
PRPQ	Programming Request for Price Quotation
TPST	Timing and Paging Statistics
WSF	Workstation Function

Index

Special Characters

Numerics

A

D

E

F

P

Q

R

S

V

W

X

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
AS/400 Performance Management Version 3 Release 1
April 1995

Publication No. GG24-3723-02

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

____

____

What other topics would you like to see in this Bulletin?

____

____

What other Technical Bulletins would you like to see published?

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

______________________________ ______________________________
Name Address

______________________________ ______________________________
Company or Organization

______________________________ ______________________________
Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 977
3605 HIGHWAY 52 NORTH
ROCHESTER MN
USA 55901-7829

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

GG24-3723-02