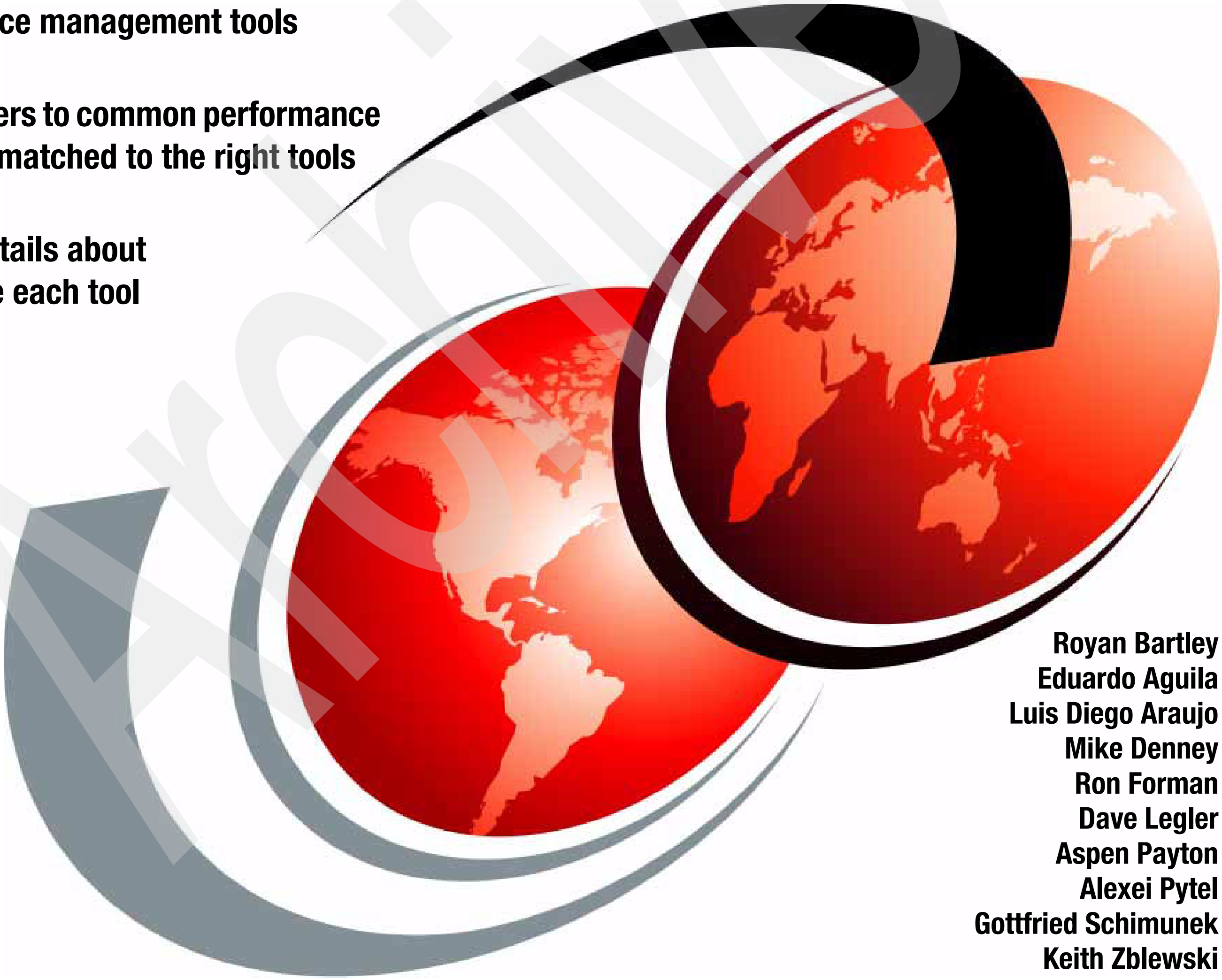
IBM @server iSeries Performance Management Tools

Learn about the various iSeries performance management tools

Find answers to common performance questions matched to the right tools

Explore details about why to use each tool



Royan Bartley
Eduardo Aguila
Luis Diego Araujo
Mike Denney
Ron Forman
Dave Legler
Aspen Payton
Alexei Pytel
Gottfried Schimunek
Keith Zblewski

Redpaper

ibm.com/redbooks

International Technical Support Organization

IBM @server iSeries Performance Management Tools

October 2005

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (October 2005)

This edition applies to IBM i5/OSVersion 5, Release 3, Modification 0 of IBM @server iSeries performance management tools.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Presentation™
AFP™
AIX®
AS/400®
Domino®
DB2®
@server®
@server®
eServer™
IBM®
i5/OS™
iSeries™
Lotus®
OS/400®
POWER™
POWER5™
Redbooks (logo) ™
Redbooks™
Tivoli®
Virtualization Engine™
WebSphere®

The following terms are trademarks of other companies:

Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft Internet Explorer, Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Preface

Learn about the complete array of IBM® @server iSeries™ performance management tools! This IBM Redpaper is designed to help you understand the different iSeries performance management tools at the IBM i5/OS™ V5R3M0 level, that are available to you and when to use them. The paper is designed for beginners who want to learn about the tools that are available. It also provides a helpful reference to the more experienced user who needs one resource for all the performance management tools available to them.

The paper begins with a comprehensive list of the tools. Then the paper presents common questions that clients ask when managing performance. Each question is matched with the recommended tool or tools to use to help you determine the answer. Finally the paper provides details about why you should use a particular tool and links to more information about obtaining and using each one.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working for the International Technical Support Organization (ITSO), Rochester Center.

Royan Bartley is an IBM Staff Software Engineer in Rockport, Maine. She has 11 years of experience with performance management tools. Her area of expertise includes PM iSeries. She holds a degree in Management Information Systems from the University of Connecticut. In addition, she earned Project Management Professional certification from the Project Management Institute.

Eduardo Aguila is an Advisory IT Specialist in IBM AMS México. He has nine years of experience in IBM @server iSeries performance tools. His areas of expertise include Performance Tools for iSeries and iSeries applications development. He holds a degree in computer science from the University of Guadalajara.

Luis Diego Araujo is an IT Specialist Advisory in Guadalajara, Jalisco, Mexico. He has eight years of experience with performance management tools. His area of expertise includes Performance Tools for iSeries. He holds a Bachelor of Science degree in Computer Engineering from the University of Guadalajara.

Mike Denney is an Advisory Software Engineer in Rochester, Minnesota. He has 23 years of experience with performance analysis, performance management tools, and work management. His areas of expertise include Performance Explorer and batch capacity planning. He holds a degree in computer science from Iowa State University. He previously published an article on how to analyze page faults.

Ron Forman is an Advisor Software Engineer for IBM in Rochester. He has 12 years of experience with Performance Monitor & Collection Services. His area of expertise is in Collection Services. He holds a degree in mathematics from the University of Nebraska at Omaha.

Dave Legler is a Staff Software Engineer for IBM in Rochester. He has 23 years of experience with performance management tools. His areas of expertise include Collection Services and Performance Explorer. He holds both a Bachelor of Science degree and a Master of Science degree in Computer Science from Michigan Technological University.

Aspen Payton is a Software Engineer for IBM in Rochester. She has six years of experience programming performance management tools. Her areas of expertise include iSeries Collection Services and iSeries iDoctor Job Watcher. She holds a degree in Management Information Systems from the University of Alaska at Anchorage.

Alexei Pytel is a Staff Software Engineer for IBM in Rochester, since 1999. He has worked with the AS/400® and iSeries since 1990 and has years of experience with performance analysis and work management. He holds a degree in Computer Science from the Institute of Physical Engineering in Moscow, Russia. He co-authored a redbook about parallel features in the OS/400® database.

Gottfried Schimunek is a Senior IT Architect with IBM in Rochester. He has had a lead role in application and product development projects for over 25 years. Currently he is the Program Manager for i5/OS and AIX® on i5, on the eServer™ Solutions Enablement team. His primary interests are performance measurement, analysis, and capacity planning of applications. Gottfried is a frequent presenter at customer, user group, and technical conferences around the world. He helps IBM Business Partners and independent software vendors (ISVs) understand how to bring the power of On Demand Business to build a secure, trusted set of applications to improve their revenue, avoid costs, and improve customer services. His particular emphasis is on the usage of existing application systems in an On Demand Business environment.

Keith Zblewski is a performance analyst and software architect in the IBM POWER™ Software Performance organization at IBM in Rochester. He joined IBM in 1987 and has been involved with various aspects of iSeries performance since 1997. He currently leads the development of sizing and capacity planning guidelines for IBM solutions on i5/OS, AIX, and Linux®. He also provides technical guidance to IBM and Business Partner sales and support teams in the area of performance and capacity planning on iSeries. Keith is a frequent speaker at user group and IBM technical conferences around the world and has published several articles on iSeries performance.

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

Jim Cook
Tom Gray
ITSO, Rochester Center

Neal Bartley
Adam Clark
Larry Cravens
Thomas Edgerton
Charles Farrell
Darrin Herrera
Donald Hoerle
Todd Kelsey
Jay Kurtz
Dawn May
Carl Obert
Lloyd Perera
Brandon Rau
Nathan Skalsky
Kurt Thompson
IBM Rochester

James Cioffi
IBM Dallas

Roderick Kagy
IBM Los Angeles

Van Sammons
IBM Atlanta

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829

1

Overview of iSeries performance management tools

When learning about the iSeries performance management tools, you need to understand what the tools are and why you would use a particular tool. In a way, it is similar to learning to drive a car and trying to understand the purpose of the speedometer compared to the odometer, even though they are based on the same data.

This redpaper covers the iSeries performance management tools and presents the information you need to guide you toward using the appropriate tool. To help understand what the tools are and when to use them, you should have a list of them available to you. This chapter provides that list to you.

Naturally you will have questions. When managing performance, asking the right questions can help to guide you to the appropriate tool. Going back to our example about learning to drive a car, you might ask, “Which tool shows total mileage of the vehicle? Which tool shows how fast the car is going?”

With performance management, you may ask the questions listed in Chapter 2, “Which tools to use” on page 5. Although every possible question is not listed in this chapter, you will find the most common ones as provided by IBM experts, IBM Business Partners, and clients who have some experience in using the tools. The purpose of the questions is to point you in the right direction to use the appropriate tool or tools. This chapter lists multiple ways, ranging from simple to complex, to find the answer.

Finally, Chapter 3, “Available iSeries performance tools” on page 21, provides the necessary details to help you understand when to use each tool or why to use one tool over another. For example, when learning to drive a car, you would learn that the purpose of the speedometer is to indicate speed and that the odometer measures the car’s mileage. This chapter provides such useful details and more about the available iSeries performance management tools.

The iSeries performance management tools presented in this paper include:

- **ANZPFRDTA**: Analyzes data from Collection Services and makes recommendations for improving performance
- **BCHMDL (Batch Model)**: Capacity planner for batch window analysis using Collection Services data
- **Collection Services**: Collects system, job, and device level statistics at regularly scheduled time intervals
- **Database Monitor (STRDBMON, ENDDBMON)**: Collects database performance statistics
- **DMPMEMINF**: Dumps information about pages of main memory to a file
- **DSPJOBTBL**: Shows the size of system job tables and the number of active jobs
- **DSPLOG**: Shows the system history log (QHST)
- **DSPPFRDTA**: Displays data from Collection Services
- **Enterprise Workload Manager (EWLM) and ARM APIs**: Monitor performance of distributed e-business applications, compare real performance-to-performance goals set in a policy, and manage domain resources to make sure that performance goals are achieved
- **IBM Director**: A cross-platform, client-based system management facility
- **iDoctor Heap analyzer for Java™**: A set of tools useful for investigating performance problems related to Java memory
- **iDoctor Job Watcher**: Shows detailed job statistics graphically and in real time
- **iDoctor PEX Analyzer**: Summarizes and analyzes Performance Explorer (PEX) data
- **iDoctor**: Suite of performance tools
- **iSeries Navigator System Monitors**: Gather and present real-time performance data that helps monitor the health of your system and identify potential performance problems before they become serious issues
- **NETSTAT and WRKTCPSTS**: Work with and display different types of TCP/IP and IPv6 network information
- **Performance Adjuster**: Automatically manages the shared storage pools without user interaction
- **Performance Explorer**: A detailed performance sampling, profiling, and trace facility
- **Performance Tools for iSeries LPP**: A set of tools for viewing, analyzing, reporting, and graphing performance data collected by Collection Services
- **Performance Tools plug-in for iSeries Navigator**: Graphical view of Collection Services data
- **PM iSeries**: Provides trend analysis based on Collection Services data
- **PRTACTRPT**: Prints a report of data collected by WRKSYSACT OUTPUT(*FILE)
- **PRTCPTRPT**: Prints a detailed device and job level view of Collection Services data
- **PRTJOBRPT**: Prints an interval-based report of jobs from Collection Services data
- **PRTLCKRPT**: Prints a report of seizes and locks from performance trace data
- **PRTPOLRPT**: Prints reports of subsystem activity and workload activity of storage pools from Collection Services data
- **PRTRSCRPT**: Prints a report of device resource usage from Collection Services data
- **PRTSYSRPT**: Prints a system view of Collection Services data

- **PRTTNSRPT**: Prints a report of performance trace data
- **PTDV**: PEX Performance Trace Data Visualizer
- **STRPFRTRC/ENDPFRTRC**: A special type of trace data customized for several reports in the Performance Tools Licensed Program Product
- **STRTRC/ENDTRC**: Collects and prints trace level data from jobs running OPM and ILE programs
- **SQL Performance Monitor**: Tracks the resources used by Structured Query Language (SQL) statements
- **WebSphere® Performance Monitoring Infrastructure (PMI)**: A framework in WebSphere Application Server that gathers performance data from various run-time resources
- **Workload Estimator (WLE)**: iSeries Capacity Planner
- **WRKACTJOB**: Shows the status and performance statistics for all active jobs in the system
- **WRKDSKSTS**: Shows the status and performance statistics for all disk devices on the system
- **WRKJOB**: Displays attributes of any job on the system
- **WRKJOBLCK**: Shows the locks on an object, the job holding each lock, and the type of the lock
- **WRKSHRPOOL**: Shows, prints, and changes shared storage pools
- **WRKSYSACT**: Shows status and performance statistics for all jobs and tasks in the system
- **WRKSYSSTS**: Shows status and performance statistics at a system level

2

Which tools to use

This chapter provides a list of performance questions that are commonly asked when working with iSeries performance management. The questions, as they are presented in this chapter, serve as a guide for those questions that are most frequently asked to help you find the most convenient tool or tools for your particular need.

For many of the questions, there are multiple ways to find the answer, so we have listed the simplest ways first, followed by more complex methods, although not every possible method is listed. The questions are grouped into logical categories for ease of use.

2.1 General system status questions

The following sections address general system performance questions.

2.1.1 Overall performance

In regard to general performance, you may find yourself asking the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- Which tool can help determine if performance is good or bad?

  What is considered acceptable performance to one person, may be unacceptable to another. Tools can show you many metrics, but only end users can decide if performance is acceptable based on throughput and response time.

- Which tool do I use to see a system view of how my system is running?

  - Use the WRKSYSSTS command. See 3.2, "Work with System Status (WRKSYSSTS) command" on page 25.
  - Use the WRKDSKSTS command. See 3.3, "Work with Disk Status (WRKDSKSTS) command" on page 26.
  - Use System Monitors in iSeries Navigator. See 3.22, "iSeries Navigator system monitors" on page 53.
  - Use the WRKACTJOB command. See 3.1, "Work with Active Jobs (WRKACTJOB) command" on page 22.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, "Performance Tools plug-in (Display Performance Data GUI)" on page 47.

- Should I collect data even if I don't have a performance problem?

  Yes, it is a good idea to always collect *STANDARDP data from Collection Services so it will be available for analysis. See 3.10, "Collection Services" on page 34.

  PM iSeries will help you automate the process of collecting data with Collection Services and with trend analysis. For more information, see 3.11, "Performance Management iSeries (PM iSeries)" on page 38.

- Which tool helps me analyze the trends in my data?

  Use PM iSeries which is explained further in 3.11, "Performance Management iSeries (PM iSeries)" on page 38.

- Where can I find system performance guidelines?

  First, refer to the Performance Capabilities Reference at the following Web site for the performance guidelines.

  `http://www-03.ibm.com/servers/eserver/iseries/perfmgmt/resource.html`

  Then use the ANZPFRDTA tool. It helps determine if your system is running within the guidelines that are stated in the Performance Capabilities Reference. See 3.16, "Performance Tools for iSeries LPP (57XX-PT1)" on page 44, to learn more information.

2.1.2 CPU usage

For CPU usage, you may find yourself asking the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- Which jobs are using the most CPU?
  - Use the WRKACTJOB command. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47, for more information.
  - Use the CPU utilization metrics from System Monitors in iSeries Navigator. See 3.22, “iSeries Navigator system monitors” on page 53.
  - Review the Job Workload Activity section of the PRTCPTRPT tool. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- How much CPU does each program use?

  Use PEX *STATS or *PROFILE, both of which are explained in 3.26, “Performance Explorer” on page 61.
- Why is job X using more CPU than it used to use?
  - Compare current performance data to previously collected data. Look for significant changes. Ensure that this data is collected with consistent characteristics. Suggested tools include:
    - Collection Services: See 3.10, “Collection Services” on page 34.
    - Performance Tools Licensed Program Product (LPP): See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
    - PEX *STATS: See 3.26, “Performance Explorer” on page 61.
    - iDoctor Job Watcher: See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
    - iDoctor PEX Analyzer: See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
    - Database Monitor: See 3.29, “Database Monitor for iSeries” on page 64.
  - Run Collection Services data collection regularly, especially prior to adding an application or additional workload. This gives you something to compare to afterwards, to help you determine the effects and then helps to pinpoint the cause. Refer to 3.10, “Collection Services” on page 34, for more information.
  - WebSphere PMI is an additional tool to use, if you are running the WebSphere application. See 3.21, “WebSphere tools” on page 53.

2.1.3 Jobs and threads

In regard to jobs and threads, you may ask the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- How many jobs are currently active?
  - Use the WRKACTJOB command. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Using the WRKSYSSTS command, specify the `ASTLVL(*BASIC)` parameter. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - From the DSPJOBTBL display, press F11 to access the in-use entries. See 3.6, “Display Job Tables (DSPJOBTBL) command” on page 32.
- Is job X multi-threaded?
  - Using the WRKJOB command, choose option 20 (Work with Threads). See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - From the WRKACTJOB command, press F11 twice to display thread data. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Using the PRTCPTRPT tool, specify the `DETAIL(*THREAD)` parameter. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- How many threads currently exist and how many are consuming CPU?
  - From the WRKACTJOB command, use option 12 (Work with Threads). See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Using the PRTCPTRPT tool, specify the `DETAIL(*THREAD)` parameter. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Which thread in a muti-threaded job uses the most resources?
  - From the WRKACTJOB command, choose option 12 (Work with Threads). See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - From the WRKJOB command, select option 20 (Work with Threads). See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - Using the PRTCPTRPT tool, specify the `DETAIL(*THREAD)` parameter. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Why is job X taking so long to complete?
  - Use the WRKACTJOB command. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Query the QAPMJOBWT database file in Collection Services. See 3.10, “Collection Services” on page 34.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.

- Does job X have contention? If so, what type of contention?
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Check the Status column in the WRKACTJOB display. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22
  - Using the WRKJOB command, select option 19 (Work with Mutexes) or option 12 (Work with Locks). See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - Use the DSPPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMJOBWT database file from Collection Services. See 3.10, “Collection Services” on page 34.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
  - Using the PRTTNSRPT command, specify the *SUMMARY report type. See 3.25, “Performance Trace Data” on page 61.
- If there is lock contention, which job is holding the lock?
  - Using the WRKJOB command, select option 12 (Work with Locks). See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - Use the WRKOBJLCK command. See 3.8, “Work with Object Locks (WRKOBJLCK) command” on page 33.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
  - Use the PRTLCKRPT command. See 3.25, “Performance Trace Data” on page 61.
- Which Structured Query Language (SQL) statements are being processed in job X?
  - In iSeries Navigator, use the SQL Performance Monitors or Database Monitor. See 3.29, “Database Monitor for iSeries” on page 64.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use PEX (for SQL server jobs only). On the ADDPEXDFN command, specify `TYPE(*TRACE) OSEVT(*DBSVRREQ)`. See 3.26, “Performance Explorer” on page 61.
- Does the system have a significant number of jobs that are starting and ending?
  - Using the DSPLOG command, look for the messages `CPF1124 Job started` and `CPF1164 Job ended`.
  - Query the QAPMJOBOS database file in Collection Services. See 3.10, “Collection Services” on page 34.
  - Query the iDoctor Job Watcher database files. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Using the PEX tool, add a PEX definition with a Type parameter of *TRACE and Base Events of *PRCCRT and *PRCDLT. See 3.26, “Performance Explorer” on page 61.
- What are a job’s run/wait characteristics over time?
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use the DSPPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.

- What is the call stack for job X over time?
  - Using the WRKJOB command, select option 11 for initial thread. See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - Using the WRKJOB command, select option 20 (Work with Threads) and then option 10 for secondary threads. See 3.7, “Work with Job (WRKJOB) command” on page 33.
  - Use the STRTRC command. See 3.15, “Start Trace (STRTRC) and End Trace (ENDTRC) commands” on page 43.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use PTDV. See 3.28, “Performance Trace Data Visualizer” on page 63.
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*TRACE) TRCTYPE(*CALLRTN)`. See 3.26, “Performance Explorer” on page 61.
- Which jobs or threads are causing my disk I/O activity?
  - Using the WRKACTJOB command, press F11 for initial thread only. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTJOBRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
- Why is job X is running longer than it used to?
  - Compare current performance data to previously collected data. Look for significant changes. Ensure this data is collected with consistent characteristics. The suggested tools include the following list.
    - Use Collection Services. See 3.10, “Collection Services” on page 34.
    - Use Performance Tools LPP. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
    - Use PEX. On the ADDPEXDFN command, specify `TYPE(*STATS)`. See 3.26, “Performance Explorer” on page 61.
    - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
    - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
    - Use Database Monitor. See 3.29, “Database Monitor for iSeries” on page 64.
  - Run Collection Services data collection regularly, especially prior to adding an application or additional workload. This gives you something to compare to afterwards, to help you determine the effects and then help pinpoint the cause. See 3.10, “Collection Services” on page 34.
  - WebSphere PMI is an additional tool to use, if you are running the WebSphere application. See 3.21, “WebSphere tools” on page 53.

2.1.4 Interactive capacity

For interactive capacity, you may ask the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- How much interactive capacity is the system using?
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use System Monitors in iSeries Navigator. See 3.22, “iSeries Navigator system monitors” on page 53.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Which jobs and threads are using the interactive capacity?
  - Using the WRKACTJOB command, look for job type INT. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Query the QAPMJOBMI database file (JBSVIF field) from Collection Services. See 3.10, “Collection Services” on page 34.

2.1.5 Logical partitioning configuration

In regard to logical partitioning configuration, you may ask the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- How many processor units does the partition have?
  - Use the WRKSYSACT (dynamic value). See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command (the value at beginning of collection). See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use the PRTSYSRPT command (the value at beginning of collection). See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command (the value at beginning of collection). See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- How many virtual processors does the partition have?
  - Use the WRKSYSACT command (dynamic value). See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command (the value at beginning of collection). See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.

- Use the PRTSYSRPT command (shows the value at the beginning of the collection). See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Use the PRTCPTRPT command (shows the value at the beginning of the collection). See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.

- How does the partition configuration change over time with dynamic logical partition (LPAR)?
  - Write the WRKSYSACT command output to an outfile and use the PRTACTRPT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Query the QAPMSYSTEM database file from Collection Services. See 3.10, “Collection Services” on page 34.
- Is the partition capped or uncapped?
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMCONF database file in Collection Services. See 3.10, “Collection Services” on page 34.
- For an uncapped partition, has the partition capacity been exceeded?

  Note: CPU% busy will be greater than 100.

  - Use the WRKSYSSTS command. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Is the partition using a shared processor pool?
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMCONF database file from Collection Services. See 3.10, “Collection Services” on page 34.
- For an uncapped partition, how much CPU is available beyond partition capacity?

  Note: This is available if the partition is authorized to see the shared processor pool data.

  - Use the Performance Tools plug-in for iSeries Navigator. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.

  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMSYSTEM database file from Collection Services. See 3.10, “Collection Services” on page 34.
- What is the utilization of the shared processor pool?

  > **Note:** This is available if the partition is authorized to see the shared processor pool data.

  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMSYSTEM database file created by Query Collection Services. See 3.10, “Collection Services” on page 34, for details.
- What is the interactive capacity of this partition?
  - Query the QAPMCONF database file created by Collection Services. See 3.10, “Collection Services” on page 34.
  - Query the QAPMSYSTEM database file created by Collection Services. See 3.10, “Collection Services” on page 34.

2.1.6 Decreased system performance

When your system experiences decreased system performance, you may ask the following questions. For each of these questions, follow the suggested guidance to the appropriate tool or information as indicated.

- Why is the system running slowly?

  > **Note:** This is a complicated question. You need to look at data from multiple tools. The following list contains possible tools to use.

  - Use the WRKDSKSTS command. See 3.3, “Work with Disk Status (WRKDSKSTS) command” on page 26.
  - Use the WRKSYSSTS command. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - Use the WRKACTJOB command. See 3.1, “Work with Active Jobs (WRKACTJOB) command” on page 22.
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.

- Why has performance become worse after adding an application, upgrading the operating system, and installing PTFs?
  - Compare current performance data to previously collected data. Look for significant changes. Ensure this data is collected with consistent characteristics. Suggested tools include:
    - Use Collection Services. See 3.10, “Collection Services” on page 34.
    - Use Performance Tools LPP. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
    - Use PEX. On the ADDPEXDFN command, specify `TYPE(*STATS)`. See 3.26, “Performance Explorer” on page 61.
    - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
    - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
    - Use Database Monitor. See 3.29, “Database Monitor for iSeries” on page 64.
  - Run Collection Services data collection regularly, especially prior to adding an application or additional workload. This will give you something to compare to afterwards, to help you determine the effects and then help pinpoint the cause. See 3.10, “Collection Services” on page 34.
  - WebSphere PMI is an additional tool to use, if you are running the WebSphere application. See 3.21, “WebSphere tools” on page 53.
- Has the partition exceeded its interactive capacity at some point?
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMSYSTEM database file from Collection Services. See 3.10, “Collection Services” on page 34.

2.1.7 Miscellaneous

The following questions apply to miscellaneous performance items. For each question, follow the suggested guidance to the appropriate tool or information as indicated.

- What is the impact of page faulting on the system? Are storage pools configured optimally?
  - Use the WRKSYSSTS command. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - Use the ANZPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
- Where is page fault data found?
  - Use the WRKSYSSTS command. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.

  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
- Is there any significant save/restore load on the system?

  Note: Look for task names starting with LD.

  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Using PEX, specify the *TRACE option on the save/restore event, OSEVT(*SAVRST). See 3.26, “Performance Explorer” on page 61.
- What is consuming temporary storage?
  - Use the WRKSYSACT command. See 3.4, “Work with System Activity (WRKSYSACT) command” on page 28.
  - Query the QAPMJOBMI database file, from Collection Services, for the JBPGA and JBPGD fields. See 3.10, “Collection Services” on page 34.
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*TRACE) STGEVT(*CRTSEG *DLTSEG *EXDSEG *TRUNCSEG)`. See 3.26, “Performance Explorer” on page 61.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.

2.2 Disk questions

The following questions specifically relate to disk performance and the corresponding tools that are useful in analyzing disk performance.

- How are the disk drives performing?
  - Use the WRKDSKSTS command. See 3.3, “Work with Disk Status (WRKDSKSTS) command” on page 26.
  - Use the Performance Tools plug-in for iSeries Navigator or the DSPPFRDTA command. See 3.17, “Performance Tools plug-in (Display Performance Data GUI)” on page 47.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTRSCRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the ANZPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.

- Which objects are being accessed on a drive or on drives?
  - Use PEX. On the ADDPEXDFN command specify `TYPE(*TRACE) DSKEVT(*ALL)`. See 3.26, “Performance Explorer” on page 61.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
- What is the average service and response time of disk units?
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTRSCRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Query the QAPMDISK database file from Collection Services. See 3.10, “Collection Services” on page 34.
- How much data is being read from or written to disk units?
  - Use the WRKDSKSTS command. See 3.3, “Work with Disk Status (WRKDSKSTS) command” on page 26.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTRSCRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Where is my disk space going? Which job or objects are consuming disk storage?
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*TRACE) STGEVT(*CRTSEG *DLTSEG *EXDSEG *TRUNCSEG)`. See 3.26, “Performance Explorer” on page 61.
- Which jobs or threads are accessing disk X?
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*TRACE) DSKEVT(*ALL)`. See 3.26, “Performance Explorer” on page 61.
- What is the response time of individual reads and writes on disk X?
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*TRACE) DSKEVT(*ALL)`. See 3.26, “Performance Explorer” on page 61.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.

2.3 Memory questions

This section lists questions that specifically relate to memory.

- How much memory does the partition have?
  - Use the WRKSYSSTS command. See 3.2, “Work with System Status (WRKSYSSTS) command” on page 25.
  - Use the WRKSHRPOOL command. See 3.14, “Work with Shared Pools (WRKSHRPOOL) command” on page 42, and 3.13, “Performance Adjuster” on page 41.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- What are the contents of memory?

  Use the DMPMEMINF tool as explained in 3.5, “Dump Main Memory Information (DMPMEMINF) command” on page 31.

2.4 Communication lines and networking questions

The following questions guide you to answers in regard to communication lines and networking.

- Which tools report general communication line performance statistics?
  - Use System Monitors in iSeries Navigator. See 3.22, “iSeries Navigator system monitors” on page 53.
  - Use the DSPPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTRSCRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the ANZPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Which tools report TCP/IP performance statistics?
  - Use the NETSTAT or WRKTCPSTS command. See 3.9, “NETSTAT or Work with TCP/IP Status (WRKTCPSTS) command” on page 34.
  - Use the PRTSYSRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use the PRTCPTRPT command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
- Are the communication line error rates reasonable?
  - Use the ANZPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.
  - Use System Monitors in iSeries Navigator. See 3.22, “iSeries Navigator system monitors” on page 53.

2.5 Database questions

This section poses questions that relate to database.

- Which tools are available to identify database performance problems?
  - Use Database Monitor. See 3.29, “Database Monitor for iSeries” on page 64.
  - Use the SQL performance monitors in iSeries Navigator. See 3.29, “Database Monitor for iSeries” on page 64.
  - Place the job in debug mode using the STRDBG command, and watch for SQL optimizer messages in job log.
  - Use PEX. On the ADDPEXDFN command, specify `TYPE(*STATS)`, and look for the QDB*, QSQ*, and QQQ* programs. See 3.26, “Performance Explorer” on page 61.
- What is the CPU usage and disk read/write activity generated by the database function?

  Use PEX. On the ADDPEXDFN command, specify `TYPE(*STATS)`, and look for the QDB*, QSQ*, and QQQ* programs. See 3.26, “Performance Explorer” on page 61.

2.6 Application questions

For these application-related questions, follow the suggested guidance to the appropriate tool or information as indicated.

- Where does my application spend most of the time?
  - Use iDoctor Job Watcher. See 3.18, “iDoctor for iSeries Job Watcher” on page 50.
  - Use iDoctor PEX Analyzer. See 3.27, “iDoctor for iSeries PEX Analyzer” on page 63.
  - Use the PRTTNSRPT RPTTYPE(*TRSIT) command. See 3.25, “Performance Trace Data” on page 61.
  - Query the database file QAPMJOBWT from Collection Services. See 3.10, “Collection Services” on page 34.
  - Use PEX (TPROF - CPU time only). On the ADDPEXDFN command, specify `TYPE(*PROFILE) PRFTYPE(*JOB)`. See 3.26, “Performance Explorer” on page 61.
- Which tools are available to analyze Java performance?
  - Use iDoctor for iSeries Heap Analyzer for Java. See 3.19, “iDoctor for iSeries Heap Analyzer for Java” on page 51.
  - Use WebSphere PMI. See 3.21, “WebSphere tools” on page 53.
  - Use PTDV. See 3.28, “Performance Trace Data Visualizer” on page 63.
- Which tools are available to analyze Lotus® Domino® performance?

  Query the database file QAPMDOMINO from Collection Services. See 3.10, “Collection Services” on page 34.
- What tools are available on i5/OS to monitor performance of other partitions (i5/OS, AIX, or Linux)?
  - Use PM iSeries. See 3.11, “Performance Management iSeries (PM iSeries)” on page 38.
  - Query the database file QAPMLPAR from Collection Services. See 3.10, “Collection Services” on page 34.

2.7 Capacity planning questions

In regard to capacity planning, you might pose the following questions. For each one, follow the suggested guidance to the appropriate tool or information as indicated.

- Which tools perform trend analysis to show system growth?

  Use PM iSeries which is explained in 3.11, “Performance Management iSeries (PM iSeries)” on page 38.

- Which tool should I use to perform capacity planning?

  Run the Workload Estimator based on PM iSeries data which is explained in 3.12, “IBM eServer Workload Estimator” on page 39, and in 3.11, “Performance Management iSeries (PM iSeries)” on page 38.

- Which tools perform capacity planning on batch jobs?

  Use Batch Model (BCHMDL; previously BATCH400) which is explained in 3.20, “Batch Model (BCHMDL)” on page 52.

- Where do I find performance specifications for the different types of disk drives?

  Refer to the Performance Capabilities Reference document located on the Performance Management Web site.

  http://www-03.ibm.com/servers/eserver/iseries/perfmgmt/resource.html

3

Available iSeries performance tools

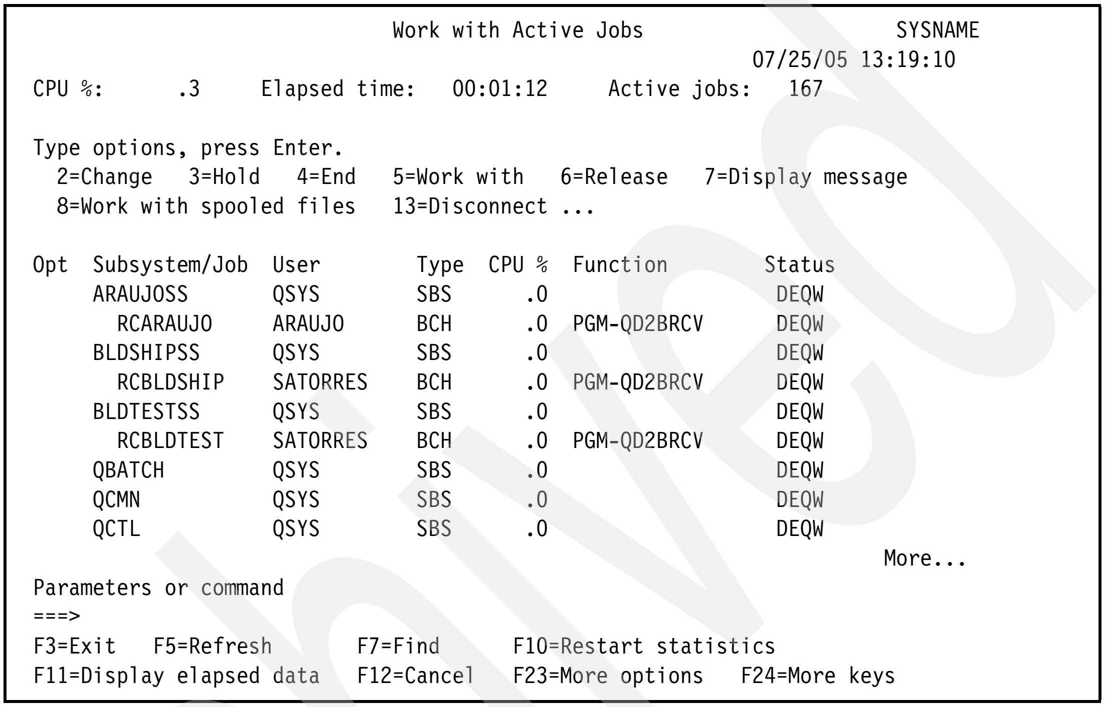
This chapter covers the iSeries performance tools that are available and why or when to use each one, as explained by the developers themselves of the tools. Although the chapter does not demonstrate how to run the tools, it provides references to other sources for you to learn how to use them.

The order of the tools as they are presented starts with the basic tools and advances to the more advanced tools. This order was chosen to help newer users who are learning about performance tools for the first time and experienced users who need a refresher on the order of complexity of each tool.

Note: The tools listed in this chapter are described with IBM i5/OS V5R3M0 function.

3.1 Work with Active Jobs (WRKACTJOB) command

You can use the WRKACTJOB command to quickly find currently active jobs that might be causing performance problems. This command is part of i5/OS. To use the command, type `WRKACTJOB` and press the Enter key. Don't worry about any of the parameters for now. You see the Work with Active Jobs display like the example in Figure 3-1.

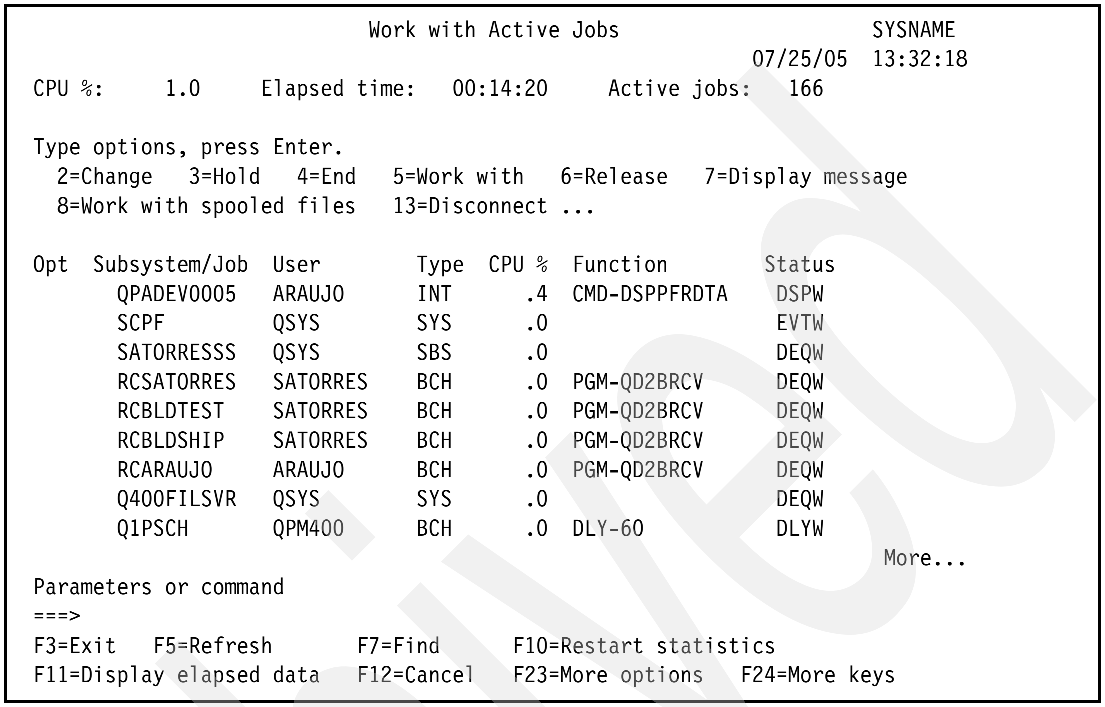


```
                              Work with Active Jobs                     SYSNAME
                                                           07/25/05 13:19:10
CPU %:      .3     Elapsed time:   00:01:12     Active jobs:   167

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     ARAUJOSS       QSYS        SBS      .0                   DEQW
       RCARAUJO     ARAUJO      BCH      .0  PGM-QD2BRCV      DEQW
     BLDSHIPSS      QSYS        SBS      .0                   DEQW
       RCBLDSHIP    SATORRES    BCH      .0  PGM-QD2BRCV      DEQW
     BLDTESTSS      QSYS        SBS      .0                   DEQW
       RCBLDTEST    SATORRES    BCH      .0  PGM-QD2BRCV      DEQW
     QBATCH         QSYS        SBS      .0                   DEQW
     QCMN           QSYS        SBS      .0                   DEQW
     QCTL           QSYS        SBS      .0                   DEQW
                                                                       More...
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 3-1 WRKACTJOB panel

Make sure the elapsed time at the top of the panel is a reasonable length of time to ensure the data is statistically meaningful. Usually the minimum is 10 seconds, but it depends on your application environment. You may want to consider 1 to 5 minutes for your time interval.

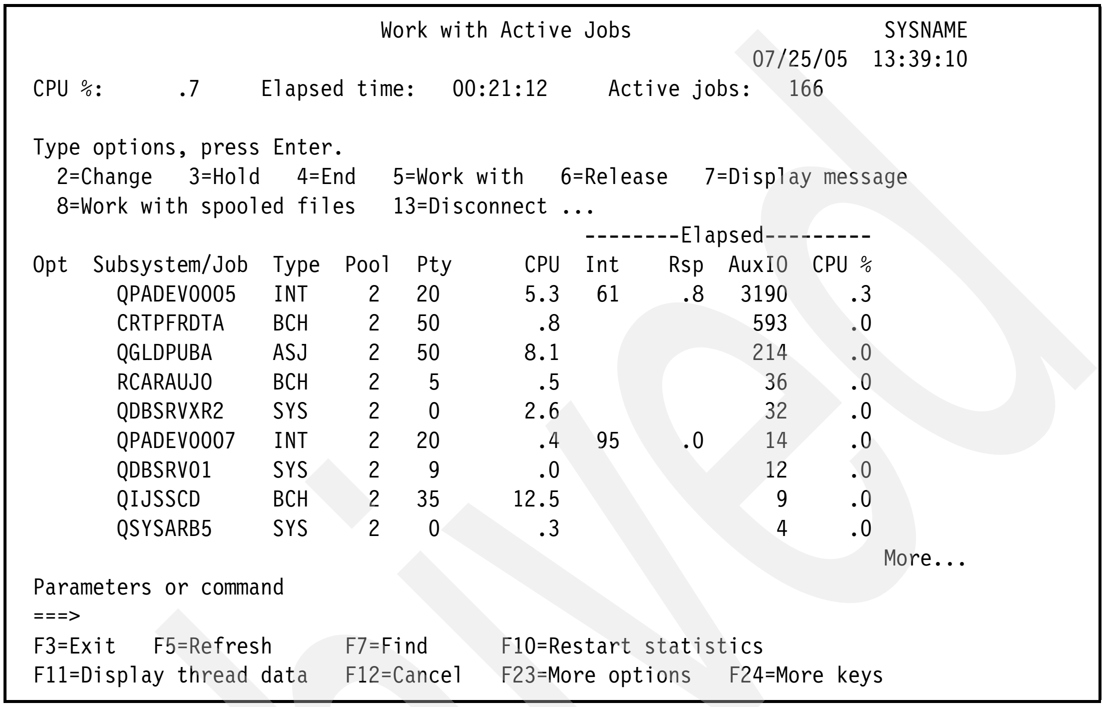
To see which jobs are using the most central processing unit (CPU) over that time interval, position the cursor over the CPU% column and press the F16 (Resequence) key. Then you see a panel similar to the one shown in Figure 3-2.



```
                              Work with Active Jobs                    SYSNAME
                                                              07/25/05  13:32:18
CPU %:     1.0     Elapsed time:   00:14:20     Active jobs:   166

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
       QPADEV0005   ARAUJO      INT      .4  CMD-DSPPFRDTA    DSPW
       SCPF         QSYS        SYS      .0                   EVTW
       SATORRESSS   QSYS        SBS      .0                   DEQW
       RCSATORRES   SATORRES    BCH      .0  PGM-QD2BRCV      DEQW
       RCBLDTEST    SATORRES    BCH      .0  PGM-QD2BRCV      DEQW
       RCBLDSHIP    SATORRES    BCH      .0  PGM-QD2BRCV      DEQW
       RCARAUJO     ARAUJO      BCH      .0  PGM-QD2BRCV      DEQW
       Q400FILSVR   QSYS        SYS      .0                   DEQW
       Q1PSCH       QPM400      BCH      .0  DLY-60           DLYW
                                                                       More...
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 3-2 WRKACTJOB sorted by CPU %

In this case, a job named QPADEV0005 is using more of the CPU than any other job or task. If you press F11 to change the view, you can position the cursor over the AuxIO column, and press F16 to sort by auxiliary input/output (I/O, disk reads and writes). You then see a panel like the one in Figure 3-3.



```
                              Work with Active Jobs                     SYSNAME
                                                             07/25/05  13:39:10
CPU %:      .7     Elapsed time:   00:21:12     Active jobs:   166

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...
                                                --------Elapsed---------
Opt  Subsystem/Job  Type  Pool  Pty      CPU  Int    Rsp  AuxIO  CPU %
       QPADEV0005   INT     2   20       5.3   61     .8   3190     .3
       CRTPFRDTA    BCH     2   50        .8                593     .0
       QGLDPUBA     ASJ     2   50       8.1                214     .0
       RCARAUJO     BCH     2    5        .5                 36     .0
       QDBSRVXR2    SYS     2    0       2.6                 32     .0
       QPADEV0007   INT     2   20        .4   95     .0     14     .0
       QDBSRV01     SYS     2    9        .0                 12     .0
       QIJSSCD      BCH     2   35      12.5                  9     .0
       QSYSARB5     SYS     2    0        .3                  4     .0
                                                                        More...
Parameters or command
===>
F3=Exit   F5=Refresh      F7=Find      F10=Restart statistics
F11=Display thread data   F12=Cancel   F23=More options   F24=More keys
```

Figure 3-3 WRKACTJOB: Using F11 to see AuxIO and F16 to sort by AuxIO

Again, the same job is at the top and is generating a lot of disk I/O. If you do not have enough disks, you may see that all jobs on the system are performing poorly. If you move the cursor to a particular column and press F16, the data is sorted based on that column. In this manner, you can isolate performance problems to a certain area. You can press F7 (Find) on large systems to quickly find an entry for a job based on the job's name, user, number, or current user. You can also search on other fields, such as subsystem, job type, status, function, pool, and priority. Again, this can help isolate problems in certain areas of the system.

If you know a job is not performing well, and you determine that neither CPU nor disk problems are happening, the job may have reached a software bottleneck (lock, seize, mutex, etc.). The status column of WRKACTJOB gives you an indication of the kind of bottleneck or wait that the job is having. If the job is in LCKW status, you can display all the locks on the job (using option 11 (Display locks)) to determine which object is being waited on.

In some cases, you may not see a job or even a group of jobs that can account for large usage of CPU or disk resources. In this case, it can be that a system task is responsible for that activity, and you need to use the WRKSYSACT command to view those tasks. The WRKSYSACT command is part of the Performance Tools Licensed Program Product (LPP).

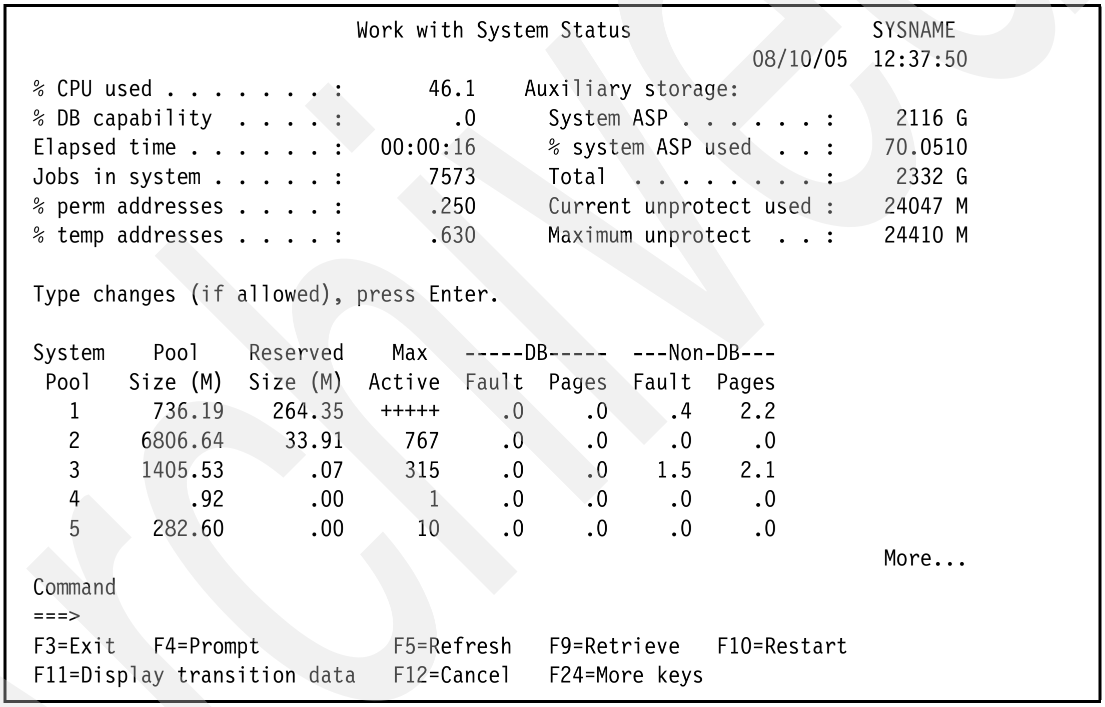
3.2 Work with System Status (WRKSYSSTS) command

The WRKSYSSTS command is part of i5/OS. The command display (Figure 3-4) shows a summary of CPU, disk, and storage pool usage. In addition, it allows memory pool sizes and activity levels to be changed. It is a quick and easy way to obtain a near real-time overview of your system's status. This command has low CPU and disk overhead associated with it, so you can run it whenever you want, without fear of degrading your system's performance.

You can update the performance data that is displayed by extending the measurement time interval and pressing F5 (Refresh). Or you can press F10 to restart the showing of the data or the statistics with the start time set to the previous display time.

The first display that you see shows the faulting rates for each pool. As a rough guide, the total faulting rate (add all of the DB Fault and Non-DB Fault values over all the pools) should be below this value:

```
100 x (% CPU used / 100) x number of processors in this partition
```



```
                             Work with System Status                    SYSNAME
                                                             08/10/05  12:37:50
% CPU used . . . . . . . :       46.1    Auxiliary storage:
% DB capability  . . . . :         .0      System ASP . . . . . . :     2116 G
Elapsed time . . . . . . :   00:00:16      % system ASP used  . . :   70.0510
Jobs in system . . . . . :       7573      Total  . . . . . . . . :     2332 G
% perm addresses . . . . :       .250      Current unprotect used :    24047 M
% temp addresses . . . . :       .630      Maximum unprotect  . . :    24410 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max    -----DB-----  ---Non-DB---
 Pool   Size (M)  Size (M)  Active  Fault  Pages  Fault  Pages
   1      736.19    264.35   +++++     .0     .0     .4    2.2
   2     6806.64     33.91     767     .0     .0     .0     .0
   3     1405.53       .07     315     .0     .0    1.5    2.1
   4         .92       .00       1     .0     .0     .0     .0
   5      282.60       .00      10     .0     .0     .0     .0
                                                                        More...
Command
===>
F3=Exit   F4=Prompt           F5=Refresh   F9=Retrieve   F10=Restart
F11=Display transition data   F12=Cancel   F24=More keys
```

Figure 3-4 WRKSYSSTS display

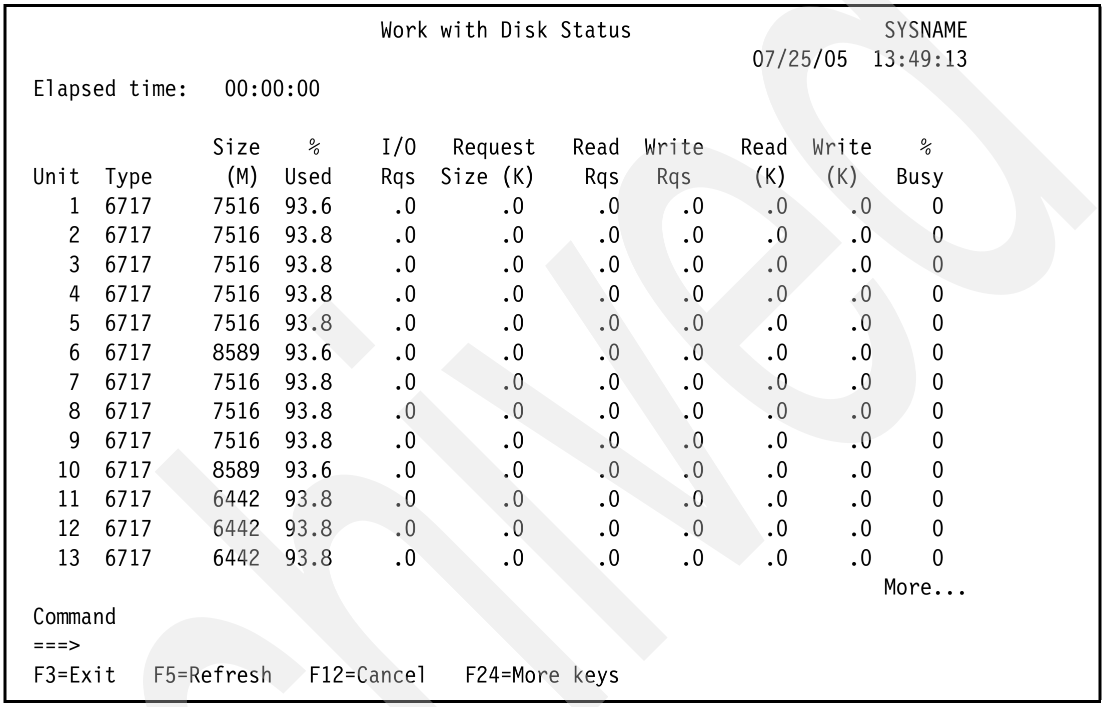
If total faulting is higher than this guideline, there are some things that you can look for. Press F11 to display state transitions. Wait->Ineligible should be close to 0, and Active->Ineligible should be 0. If neither is at the expected value, either select one of the following actions (easy to complex) or complete them all in the order shown.

1. Increase the activity level of the problem pool (Max Active).
2. Move storage around (manually, or with QPFRADJ) so the problem pool has more storage.
3. Add main storage to the system.

3.3 Work with Disk Status (WRKDSKSTS) command

You can use the WRKDSKSTS command to make a quick assessment of your system's disk configuration and check whether there is a potential performance problem with disk units. The WRKDSKSTS command is always available and is a part of i5/OS.

Using the command is easy; you simply type WRKDSKSTS on any command line. Then you see a display like the one in Figure 3-5. The display shows the list of accessible disk units and a few basic performance counters.

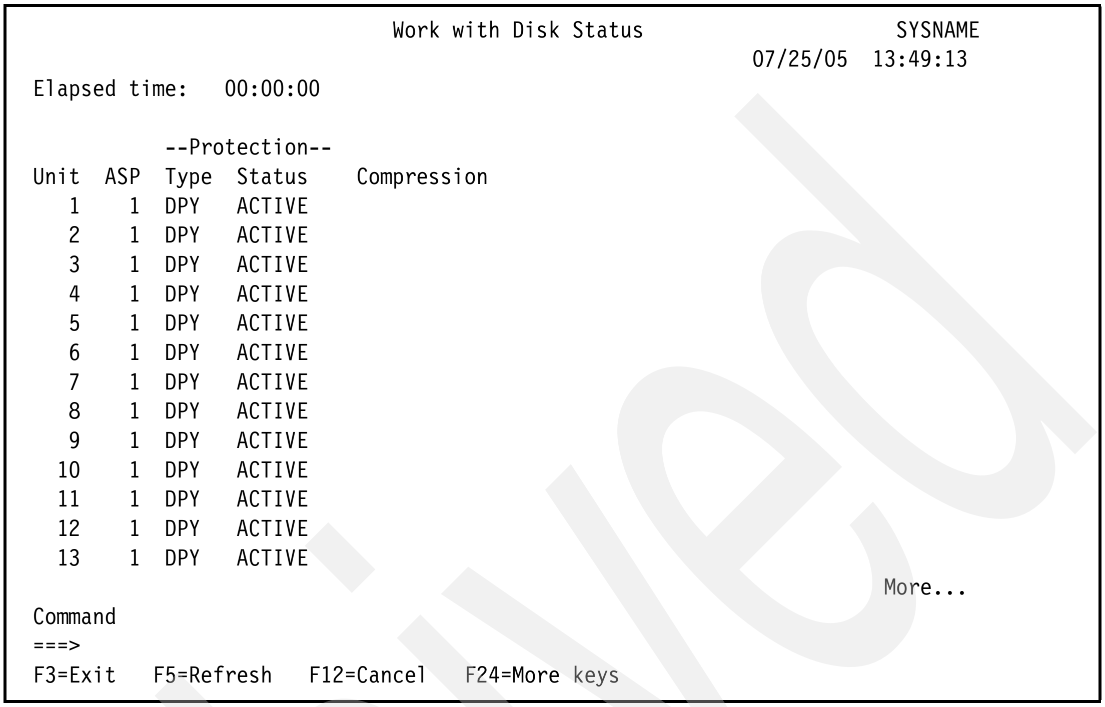


```
                              Work with Disk Status                     SYSNAME
                                                               07/25/05  13:49:13
 Elapsed time:   00:00:00

               Size    %     I/O   Request   Read  Write   Read  Write    %
 Unit  Type     (M)  Used    Rqs  Size (K)    Rqs   Rqs     (K)   (K)   Busy
    1  6717    7516  93.6     .0        .0     .0     .0     .0     .0     0
    2  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    3  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    4  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    5  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    6  6717    8589  93.6     .0        .0     .0     .0     .0     .0     0
    7  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    8  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
    9  6717    7516  93.8     .0        .0     .0     .0     .0     .0     0
   10  6717    8589  93.6     .0        .0     .0     .0     .0     .0     0
   11  6717    6442  93.8     .0        .0     .0     .0     .0     .0     0
   12  6717    6442  93.8     .0        .0     .0     .0     .0     .0     0
   13  6717    6442  93.8     .0        .0     .0     .0     .0     .0     0
                                                                          More...
 Command
 ===>
 F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 3-5 WRKDSKSTS display

All performance measurements performed by the WRKDSKSTS command apply to a certain measurement period. This period is indicated by the elapsed time, shown in the top left corner of the panel. The elapsed time must be large enough to ensure that collected performance data is statistically valid. The % Busy column is the most sensitive to the length of the measurement interval. However, it depends on the application environment. In particular, heavy disk workload can provide enough statistical data in a relatively short time, while periods of light activity may require longer measurement interval.

When you press F11 on the initial panel, you see a display that shows additional details, like the example in Figure 3-6.



```
                              Work with Disk Status                    SYSNAME
                                                              07/25/05  13:49:13
Elapsed time:   00:00:00

           --Protection--
Unit  ASP  Type  Status    Compression
   1    1  DPY   ACTIVE
   2    1  DPY   ACTIVE
   3    1  DPY   ACTIVE
   4    1  DPY   ACTIVE
   5    1  DPY   ACTIVE
   6    1  DPY   ACTIVE
   7    1  DPY   ACTIVE
   8    1  DPY   ACTIVE
   9    1  DPY   ACTIVE
  10    1  DPY   ACTIVE
  11    1  DPY   ACTIVE
  12    1  DPY   ACTIVE
  13    1  DPY   ACTIVE
                                                                         More...
Command
===>
F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 3-6 WRKDSKSTS: Using F11 to display storage use

Watch for the following items on the WRKDSKSTS display.

- Disk units with a high value in the % Busy column

  Very high utilization has adverse effect on the response time of disk I/O operations. Acceptable utilization numbers vary depending on disk type and configuration, but in general the recommended range should not exceed 50% to 70%.

- Disk units with an unusually high number of disk I/O operations

  This can be a sign of an unbalanced disk configuration.

- Disk units with significantly different sizes in the same auxiliary storage pool (ASP)

  Disk units with a higher capacity may become a performance bottleneck.

- Disk units with significantly different values in the % Used column in the same ASP

  This may happen when new disk units are added to the ASP without proper balancing.

- DEGRADED value in the Status column

  This value can indicate a hardware problem that is causing performance problems.

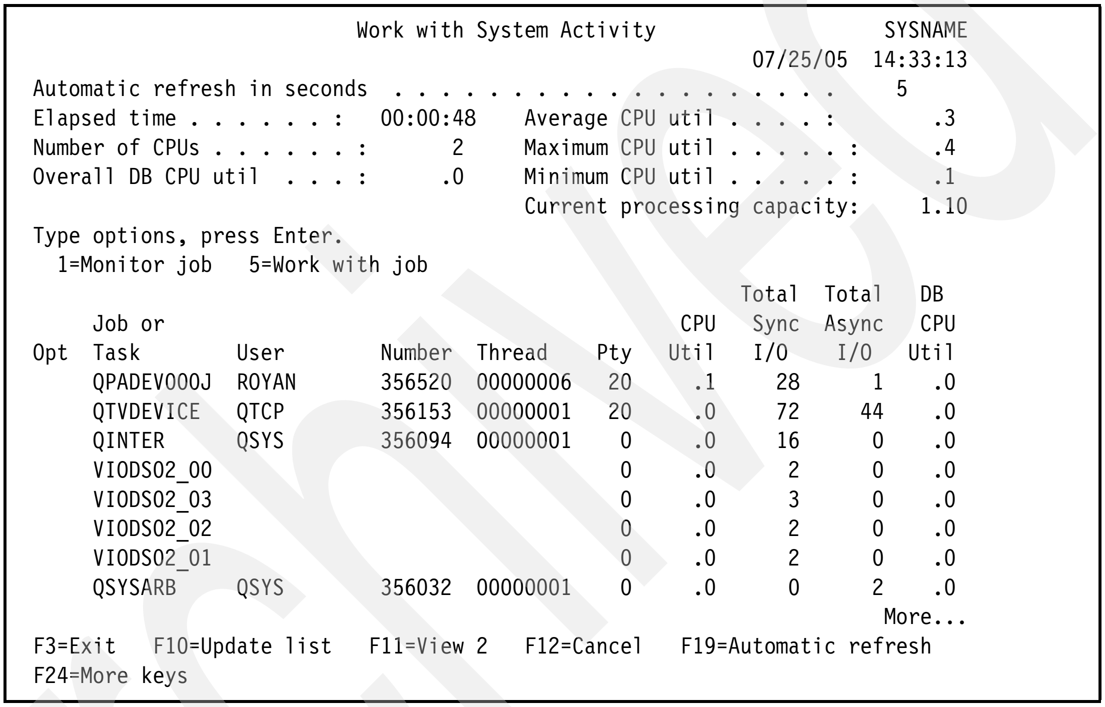
For more detailed analysis of the disk performance during extended periods of time, use the Collection Services and Performance Tools reports, which are based on the performance data collected by Collection Services.

3.4 Work with System Activity (WRKSYSACT) command

You can use the WRKSYSACT command to identify the jobs and tasks that might be causing performance problems in real time. Unlike the WRKACTJOB command, the WRKSYSACT command can display system tasks. Figure 3-7 shows a sample display.

The WRKSYSACT command is part of the Performance Tools LPP. For more information, see 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44.

Note: Only one call of the WRKSYSACT command can be active at the same time.



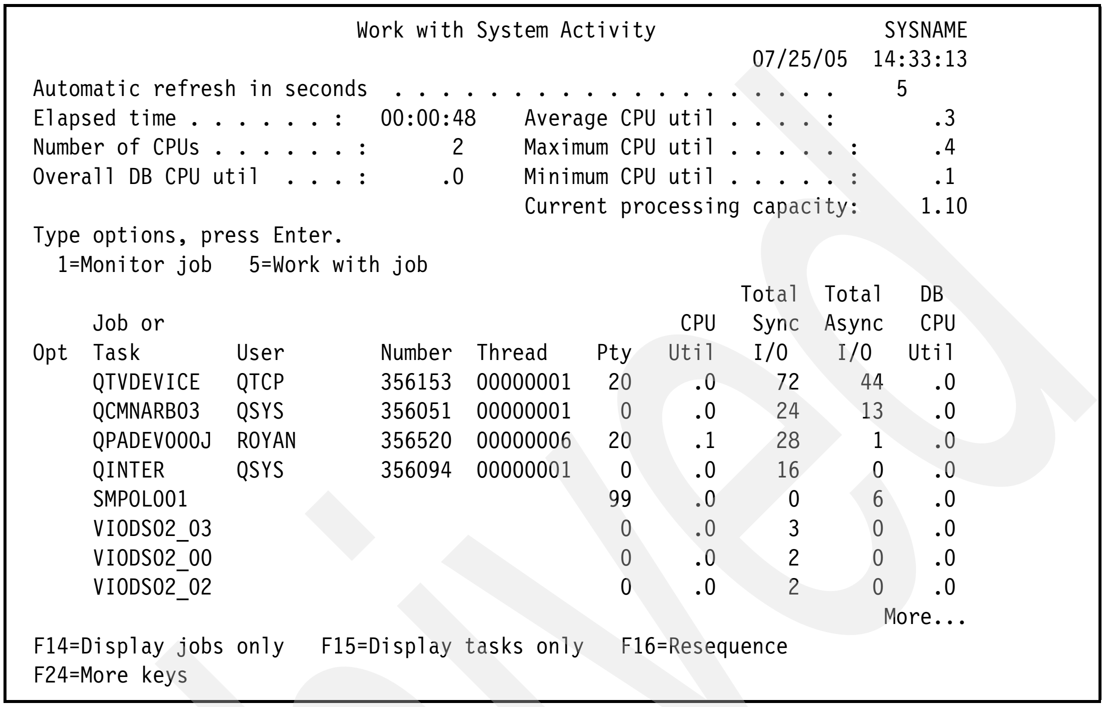
```
                           Work with System Activity                   SYSNAME
                                                            07/25/05  14:33:13
Automatic refresh in seconds  . . . . . . . . . . . . . . . . . . .     5
Elapsed time . . . . . . :   00:00:48    Average CPU util . . . . :        .3
Number of CPUs . . . . . . :       2     Maximum CPU util . . . . . :      .4
Overall DB CPU util  . . . :      .0     Minimum CPU util . . . . . :      .1
                                         Current processing capacity:     1.10
Type options, press Enter.
  1=Monitor job   5=Work with job
                                                           Total  Total   DB
     Job or                                           CPU   Sync  Async   CPU
Opt  Task        User        Number  Thread    Pty   Util   I/O    I/O   Util
     QPADEV000J  ROYAN       356520  00000006   20     .1     28      1    .0
     QTVDEVICE   QTCP        356153  00000001   20     .0     72     44    .0
     QINTER      QSYS        356094  00000001    0     .0     16      0    .0
     VIODS02_00                                  0     .0      2      0    .0
     VIODS02_03                                  0     .0      3      0    .0
     VIODS02_02                                  0     .0      2      0    .0
     VIODS02_01                                  0     .0      2      0    .0
     QSYSARB     QSYS        356032  00000001    0     .0      0      2    .0
                                                                       More...
F3=Exit   F10=Update list   F11=View 2   F12=Cancel   F19=Automatic refresh
F24=More keys
```

Figure 3-7 WRKSYSACT display

The panel shows the CPU utilization and the total number of physical disk I/O operations (synchronous and asynchronous) of all jobs and system tasks based on the elapsed time. Make sure the elapsed time is a reasonable length of time to ensure that performance data is statistically valid. It depends on your application environment, but the minimum suggested value is 5 seconds. To be displayed by the WRKSYSACT command, jobs and tasks must use at least one-tenth of 1% (.1%) of the processor or perform one disk I/O operation.

All jobs and system tasks are ordered by the amount of processing time they use during the interval. This is the default sequence for the WRKSYSACT. However, you can also sort the jobs and system tasks by I/O operations, net storage, allocated storage, deallocated storage, DB CPU, and total waiting time.

For example, to see which jobs or system tasks are performing most of the physical disk I/O operations, press F16 (Resequence) and select option 2 (Sequence by I/O). You then see a display similar to the one in Figure 3-8.

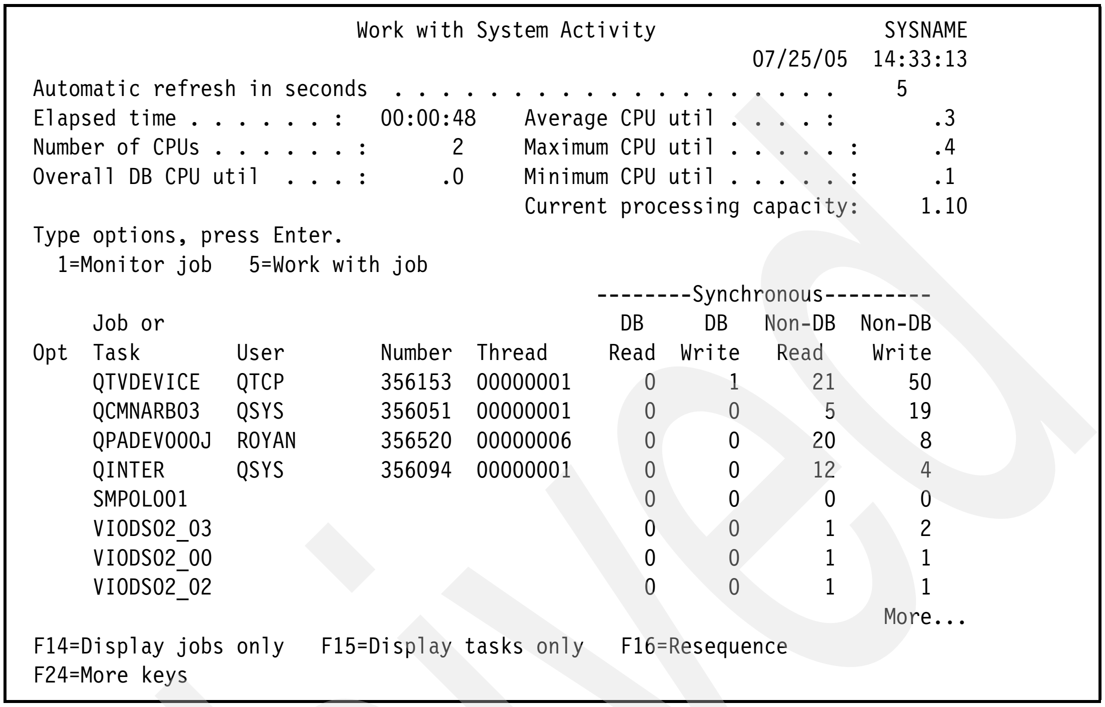


```
                             Work with System Activity                   SYSNAME
                                                                07/25/05  14:33:13
Automatic refresh in seconds  . . . . . . . . . . . . . . . . . . .     5
Elapsed time . . . . . . :   00:00:48    Average CPU util . . . . :        .3
Number of CPUs . . . . . . :       2     Maximum CPU util . . . . . :      .4
Overall DB CPU util  . . . :      .0     Minimum CPU util . . . . . :      .1
                                         Current processing capacity:     1.10
Type options, press Enter.
  1=Monitor job   5=Work with job
                                                                Total  Total   DB
     Job or                                               CPU   Sync  Async   CPU
Opt  Task        User        Number  Thread    Pty  Util   I/O    I/O  Util
     QTVDEVICE   QTCP        356153  00000001   20    .0     72     44    .0
     QCMNARB03   QSYS        356051  00000001    0    .0     24     13    .0
     QPADEV000J  ROYAN       356520  00000006   20    .1     28      1    .0
     QINTER      QSYS        356094  00000001    0    .0     16      0    .0
     SMPOL001                                   99    .0      0      6    .0
     VIODS02_03                                  0    .0      3      0    .0
     VIODS02_00                                  0    .0      2      0    .0
     VIODS02_02                                  0    .0      2      0    .0
                                                                           More...
F14=Display jobs only   F15=Display tasks only   F16=Resequence
F24=More keys
```

Figure 3-8 WRKSYSACT sequenced by I/O

The jobs that perform most of the disk I/O operations may need further investigation, if you are experiencing disk I/O problems. The system uses asynchronous disk I/O operations wherever possible to optimize performance. Asynchronous disk I/O operations do not require the job or task to wait for disk I/O operations to finish, so asynchronous disk I/Os are preferable to synchronous disk I/Os.

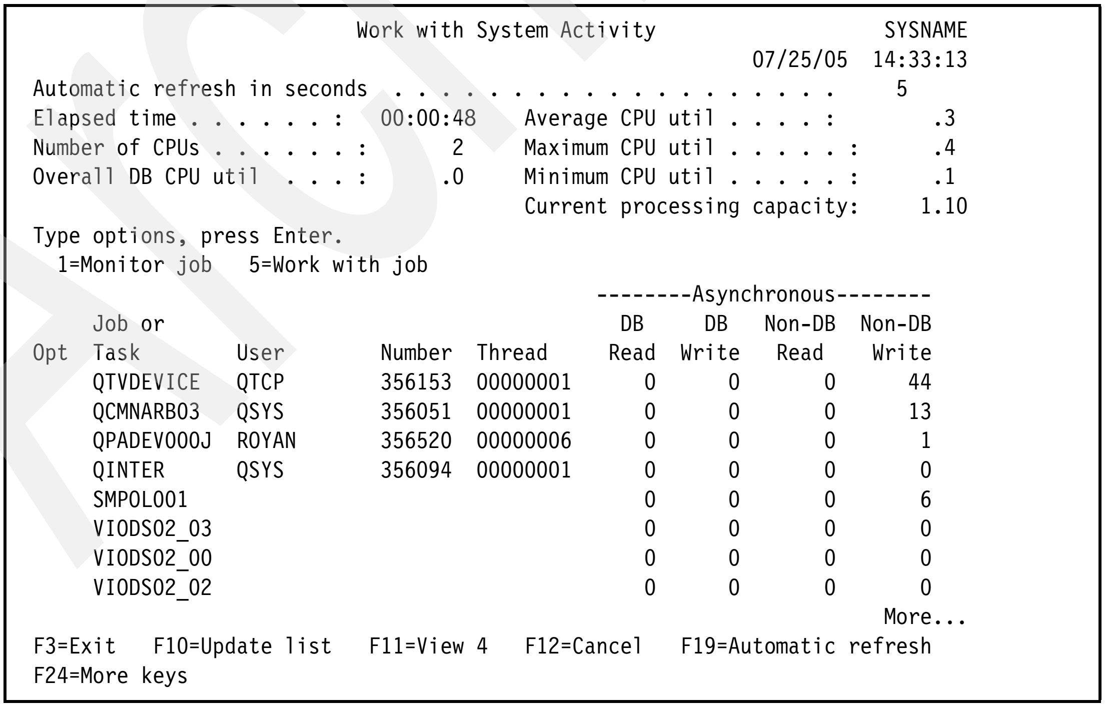
When you press F11 on the first panel to change the view, you see the synchronous disk I/O operations detail on a display like the example in Figure 3-9.



```
                             Work with System Activity                  SYSNAME
                                                                07/25/05  14:33:13
 Automatic refresh in seconds  . . . . . . . . . . . . . . . . . . .     5
 Elapsed time . . . . . . :   00:00:48    Average CPU util . . . . :       .3
 Number of CPUs . . . . . . :       2     Maximum CPU util . . . . . :      .4
 Overall DB CPU util  . . . :      .0     Minimum CPU util . . . . . :      .1
                                          Current processing capacity:     1.10
 Type options, press Enter.
   1=Monitor job   5=Work with job
                                                --------Synchronous---------
      Job or                                      DB     DB   Non-DB  Non-DB
 Opt  Task        User        Number  Thread     Read  Write   Read    Write
      QTVDEVICE   QTCP        356153  00000001      0      1      21      50
      QCMNARB03   QSYS        356051  00000001      0      0       5      19
      QPADEV000J  ROYAN       356520  00000006      0      0      20       8
      QINTER      QSYS        356094  00000001      0      0      12       4
      SMPOL001                                      0      0       0       0
      VIODS02_03                                    0      0       1       2
      VIODS02_00                                    0      0       1       1
      VIODS02_02                                    0      0       1       1
                                                                       More...
 F14=Display jobs only   F15=Display tasks only   F16=Resequence
 F24=More keys
```

Figure 3-9 WRKSYSACT: View 2 pressing F11 to see synchronous disk I/O operations detail

Now, if you press F11 key on the second panel to change the view, you see the asynchronous disk I/O operations detail like the example in Figure 3-10.



```
                             Work with System Activity                  SYSNAME
                                                                07/25/05  14:33:13
 Automatic refresh in seconds  . . . . . . . . . . . . . . . . . . .     5
 Elapsed time . . . . . . :   00:00:48    Average CPU util . . . . :       .3
 Number of CPUs . . . . . . :       2     Maximum CPU util . . . . . :      .4
 Overall DB CPU util  . . . :      .0     Minimum CPU util . . . . . :      .1
                                          Current processing capacity:     1.10
 Type options, press Enter.
   1=Monitor job   5=Work with job
                                                --------Asynchronous--------
      Job or                                      DB     DB   Non-DB  Non-DB
 Opt  Task        User        Number  Thread     Read  Write   Read    Write
      QTVDEVICE   QTCP        356153  00000001      0      0       0      44
      QCMNARB03   QSYS        356051  00000001      0      0       0      13
      QPADEV000J  ROYAN       356520  00000006      0      0       0       1
      QINTER      QSYS        356094  00000001      0      0       0       0
      SMPOL001                                      0      0       0       6
      VIODS02_03                                    0      0       0       0
      VIODS02_00                                    0      0       0       0
      VIODS02_02                                    0      0       0       0
                                                                       More...
 F3=Exit   F10=Update list   F11=View 4   F12=Cancel   F19=Automatic refresh
 F24=More keys
```

Figure 3-10 WRKSYSACT: View 3 from using F11 to see asynchronous disk I/O operations detail

Using the F16 (Resequence) key can help you isolate performance problems to a certain area.

Press F19 to use the automatic refresh feature of WRKSYSACT. It automatically updates the WRKSYSACT display after the number of seconds indicated in the Automatic refresh field (shown in the top right corner of the panel). The automatic refresh mode is one of the most useful functions to identify the jobs and system tasks that are not performing well. These jobs or tasks may be those that consume the most CPU, perform most of the physical disk I/O operations, etc.

The automatic refresh mode (F19) and the Resequence (F16) functions together become a useful feature of WRKSYSACT to identify the jobs and system tasks that are not performing well. If you see that the same jobs or system tasks are consistently at the top, you need to investigate them to determine why.

In addition to these features, you can run the WRKSYSACT command in batch mode to collect performance data for later analysis. To use this function, you need to specify the *FILE option on the OUTPUT parameter. The collected performance data is written to the QAITMON database file. You can find more detail about QAITMON in *Performance Tools for iSeries*, SC41-5340-01.

You can use the Print Activity Report (PRTACTRPT) command to process the performance data collected in QAITMON file. To learn about this command, again refer to *Performance Tools for iSeries*, SC41-5340-01.

3.5 Dump Main Memory Information (DMPMEMINF) command

> **Note:** The DMPMEMINF command is available via an English-only PTF SI12672 in V5R3. It will be part of i5/OS releases after V5R3.

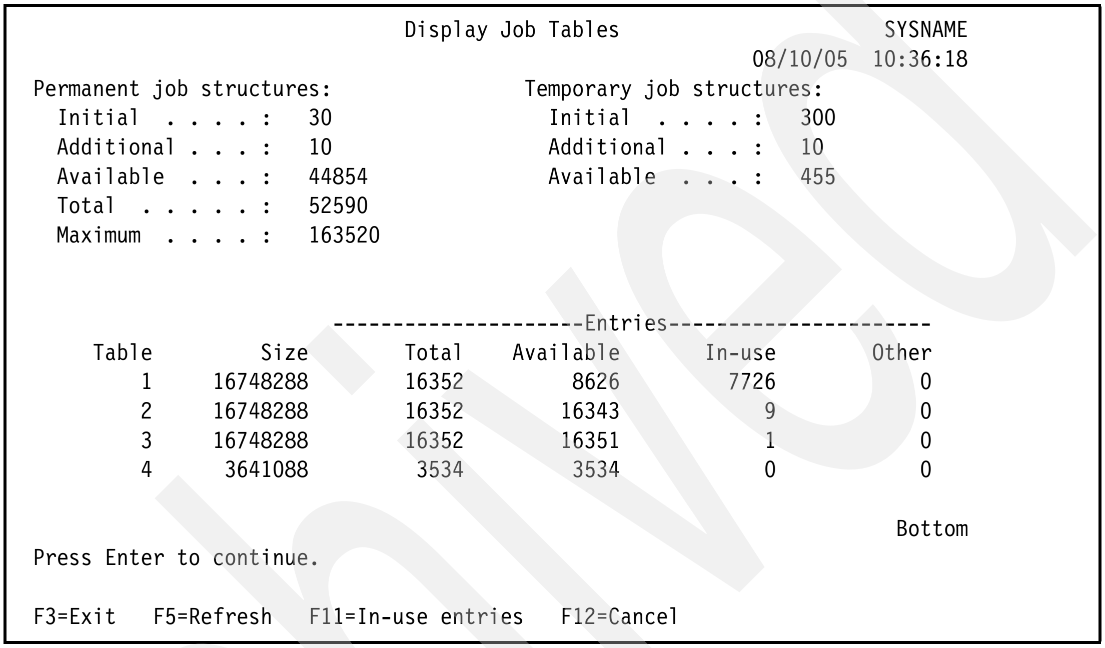
The DMPMEMINF command dumps information about the contents of memory into a database file for performance analysis. Each record in the file represents one page of memory. The NBRPAGE keyword specifies the number of pages dumped. This number of pages is selected randomly from all of the pages in memory. If you want to dump a record for every page in memory, specify NBRPAGE(*ALL), but be aware that this option can produce a very large file.

You can write queries to summarize the data in the output file in useful ways. For example, the following query tells you which object names are using the greatest number of pages. We recommend that you run this command in its own storage pool. This helps to avoid changing the main memory pages that reside in the storage pools that are being dumped.

```
DMPMEMINF OUTFILE(mylib/dmpmem)

SELECT COUNT(*),POOL, OBJNAME, LIBNAME FROM mylib/dmpmem GROUP
BY POOL, OBJNAME, LIBNAME ORDER BY 1 DESC
```

3.6 Display Job Tables (DSPJOBTBL) command

The DSPJOBTBL command is part of i5/OS. It displays information about the internal job tables used by the operating system. Compare the number of available entries to the number of in-use entries. If there are no available entries, the system will experience a performance degradation when it starts a new job because the table needs to be extended. However, if there are too many available entries, the time needed for IPL will be longer than necessary. Consider the sample output shown in Figure 3-11.



```
                                 Display Job Tables                      SYSNAME
                                                              08/10/05  10:36:18
 Permanent job structures:                 Temporary job structures:
   Initial  . . . . :   30                   Initial  . . . . :   300
   Additional . . . :   10                   Additional . . . :   10
   Available  . . . :   44854                Available  . . . :   455
   Total  . . . . . :   52590
   Maximum  . . . . :   163520



                            --------------------Entries----------------------
      Table        Size        Total    Available       In-use        Other
          1    16748288        16352         8626         7726            0
          2    16748288        16352        16343            9            0
          3    16748288        16352        16351            1            0
          4     3641088         3534         3534            0            0

                                                                      Bottom
 Press Enter to continue.

 F3=Exit   F5=Refresh   F11=In-use entries   F12=Cancel
```

Figure 3-11 DSPJOBTBL command sample output

On this system, there are four job tables, but tables 2 through 4 have only minimal use. This situation can be caused by a program which automatically creates new jobs or threads and becomes stuck in an endless loop. By the time the job is ended, thousands of new jobs could be created. When these jobs have ended, the slots allocated for them in the job table remain, but are often unused by any other jobs.

You can remove unnecessary entries in the job table during the IPL process by changing the IPL attributes using the Change IPL Attributes (CHGIPLA CPRKJOBTBL(*ALL)) command. Job tables will be compressed on the next IPL after this attribute is changed. Don't forget to change it back to *NONE after the IPL is done, or it will compress the job table after every IPL. However, we do not recommend that you use this method because compressing the job tables increases the IPL time, since every table entry must be examined.

For more information about the DSPJOBTBL command, refer to the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/cl/dspjobtbl.htm

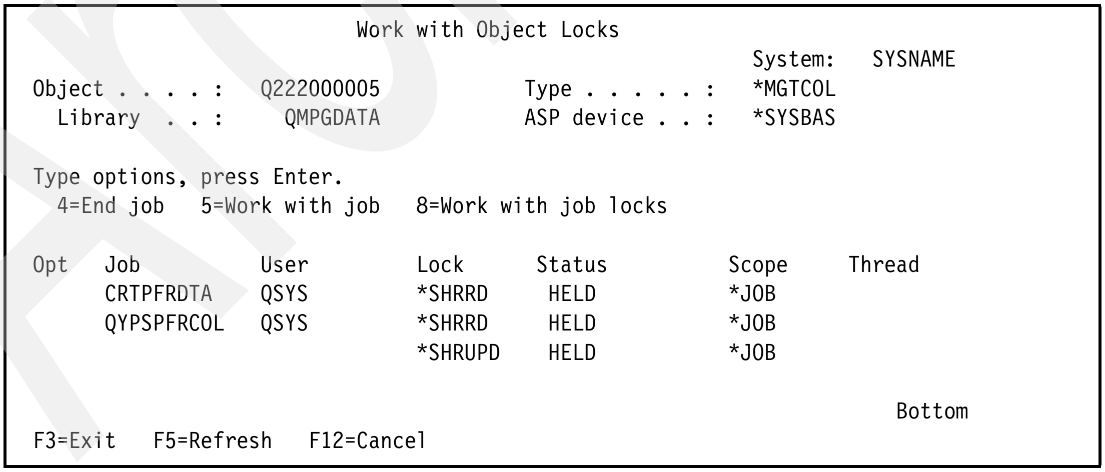
3.7 Work with Job (WRKJOB) command

The WRKJOB command is part of i5/OS. It allows you to work with or change information related to a specific job. This command supports the following options.

- 1. Display job status attributes
- 2. Display job definition attributes
- 3. Display job run attributes, if active
- 4. Work with spooled files
- 10. Display job log, if active or on job queue
- 11. Display call stack, if active
- 12. Work with locks, if active
- 13. Display library list, if active
- 14. Display open files, if active
- 15. Display file overrides, if active
- 16. Display commitment control status, if active
- 17. Display communications status, if active
- 18. Display activation groups, if active
- 19. Work with mutexes, if active
- 20. Work with threads, if active
- 21. Work with media library attributes, if active
- 30. All of the above
- 40. Change job
- 41. End job
- 42. Hold job
- 43. Release job

3.8 Work with Object Locks (WRKOBJLCK) command

The WRKOBJLCK command is part of i5/OS. It displays a list of locks against a specific object. You provide the object and library name, as well as the object type and member name if necessary. Figure 3-12 shows a sample output.

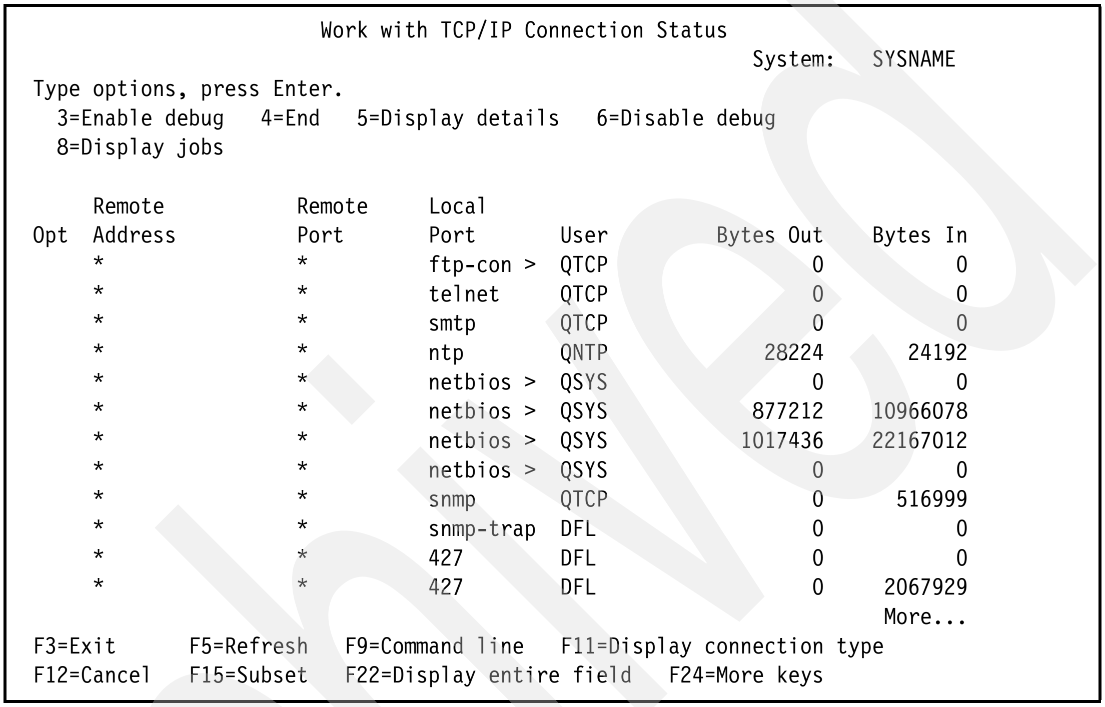


```
                             Work with Object Locks
                                                          System:   SYSNAME
Object . . . . :   Q222000005          Type . . . . . :   *MGTCOL
  Library  . . :     QMPGDATA          ASP device . . :   *SYSBAS

Type options, press Enter.
  4=End job   5=Work with job   8=Work with job locks

Opt   Job          User         Lock      Status          Scope     Thread
      CRTPFRDTA    QSYS         *SHRRD     HELD           *JOB
      QYPSPFRCOL   QSYS         *SHRRD     HELD           *JOB
                                *SHRUPD    HELD           *JOB

                                                                        Bottom
F3=Exit   F5=Refresh   F12=Cancel
```

Figure 3-12 WRKOBJLCK example

3.9 NETSTAT or Work with TCP/IP Status (WRKTCPSTS) command

This command, which is part of i5/OS, is known both as NETSTAT and WRKTCPSTS. It has various options, but option 3 (Work with TCP/IP connection status) is of interest for performance analysis. If you select this option and then press F11, you see counts for bytes in and bytes out for each remote host as shown in the example in Figure 3-13.



```
                          Work with TCP/IP Connection Status
                                                              System:   SYSNAME
 Type options, press Enter.
   3=Enable debug   4=End   5=Display details   6=Disable debug
   8=Display jobs

      Remote           Remote     Local
 Opt  Address          Port       Port       User         Bytes Out    Bytes In
      *                *          ftp-con >  QTCP                 0           0
      *                *          telnet     QTCP                 0           0
      *                *          smtp       QTCP                 0           0
      *                *          ntp        QNTP             28224       24192
      *                *          netbios >  QSYS                 0           0
      *                *          netbios >  QSYS            877212    10966078
      *                *          netbios >  QSYS           1017436    22167012
      *                *          netbios >  QSYS                 0           0
      *                *          snmp       QTCP                 0      516999
      *                *          snmp-trap  DFL                  0           0
      *                *          427        DFL                  0           0
      *                *          427        DFL                  0     2067929
                                                                        More...
 F3=Exit      F5=Refresh   F9=Command line   F11=Display connection type
 F12=Cancel   F15=Subset   F22=Display entire field   F24=More keys
```

Figure 3-13 WRKTCPSTS command sample output

3.10 Collection Services

Collection Services, a component of i5/OS, collects performance data at specific time intervals. This is referred to as *time interval data* or *sample data*. You can think of it as a snapshot of activity that has taken place up to the time that each sample was collected. Many configuration options are available to customize the data collected.

You can export this data to a set of database files referred to as the *performance database* (see the discussion about the Create Performance Data (CRTPFRDTA) command in 3.10.1, “How Collection Services works” on page 35). These files are then used by most other performance reporting and analysis tools, such as Performance Tools LPP (see 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44) and PM iSeries (see 3.11, “Performance Management iSeries (PM iSeries)” on page 38), that use time interval data as input.

Time interval data is the basis for many performance tools. It has the least impact and collects the widest set of performance data compared to other tools used to monitor a system’s performance and possibly detect an approaching problem. Tools based on time interval data are the first step in problem investigation. Time interval data usually points you to a problem area to investigate further and, at times, it may even pinpoint the problem.

By monitoring a system with time interval data over time, you can further use it to do trending (to project when more CPU, memory, or disk is needed) and capacity planning (to predict the effects of adding more workload). When something changes, you can compare performance data from the before and after period to help identify what is different.

3.10.1 How Collection Services works

Collection Services is a single system focal point for the collection of time interval data. The same data is shared by different, unique functions that require the same data as input. This means that Collection Services has two modes of operation, and at any time, either or both may be active.

- The primary (most often used) function is the *profile collection*. This can be thought of as the system collection. A system administrator configures Collection Services to automatically run, continuously collecting a general set of data, usually at 15 minute time intervals, for later analysis. This data is used by external tools for trending, capacity planning, and a first pass at problem determination. On most systems, this mode of operation is configured and automatically started by PM iSeries. For more information, see 3.11, “Performance Management iSeries (PM iSeries)” on page 38.
- The secondary function supports a smaller set of specific data that is collected more frequently, such as in 1 minute intervals, for real-time system monitoring. This is referred to as the *application collection*. The most common users of this operating mode are the Management Central system and job monitors.

Collection Services does not process the data during collection. Rather, it is stored in a raw form in a special release-independent object that is optimized for this purpose: the management collection (*MGTCOL) object. Data in the management collection object can be processed immediately at each collection interval boundary, concurrent with collection. Or processing can be deferred until the data is needed or when more system resources are available. The management collection object can be saved and restored to another system for processing regardless of release.

The CRTPFRDTA command is used to export data from the management collection object to the performance database. You can use the CRTPFRDTA command with all defaults meaning that the data will be exported exactly as collected. Or you can specify command options to export only specific data of interest. You can set up Collection Services to run the CRTPFRDTA command automatically as a batch job that exports data concurrently with the collection.

The performance database is a set of approximately 50 database files that may contain all the different types of data that Collection Services is capable of collecting. The data that you obtain (the files used) depends on the hardware and software functions running on your system, combined with the data you selected. The latter depends on both the data collected (what the management collection object contains) and what you select to export.

Within the performance database, a specific collection (data from a management collection object) is represented by a database member within each file (the same member name is used across all files). Thus the performance database can contain multiple members, each representing a different data collection.

For more information about the available files and the data (fields) within the files, refer to the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzahx/rzahxperfdatafiles1.htm

3.10.2 How to use Collection Services

You can configure and manage Collection Services in many ways.

- Commands

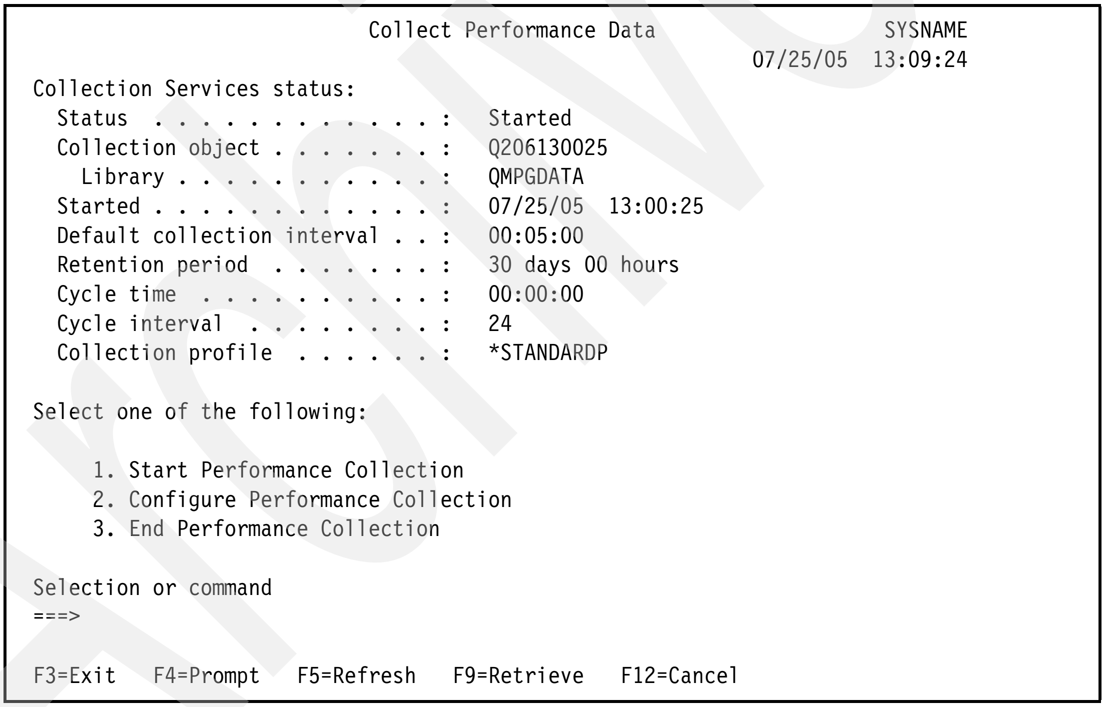
  Beginning in V5R3, commands are available to configure (CFGPFRCOL), start (STRPFRCOL), and end (ENDPFRCOL) the Collection Services performance collection. This is the easiest method. To see your current configuration options, type the CFGPFRCOL command.

- System application programming interfaces (APIs)

  System APIs are available to manage Collection Services. Use QYPSCSCA or QypsChgSysCollectorAttributes to configure the collector. Use QYPSSTRC or QypsStartCollector to start the collector. And use QYPSENDC or QypsEndCollector to end the collector. Refer to the iSeries Information Center to learn more about these APIs.

  http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/apis/perfmgmt_colsvc1.htm

- The Performance Tools LPP PERFORM menu (also the Start Performance Tools (STRPFRT) command) provides a menu interface to Collection Services. Use option 2 (Collect performance data) to configure, start, and stop the collector.

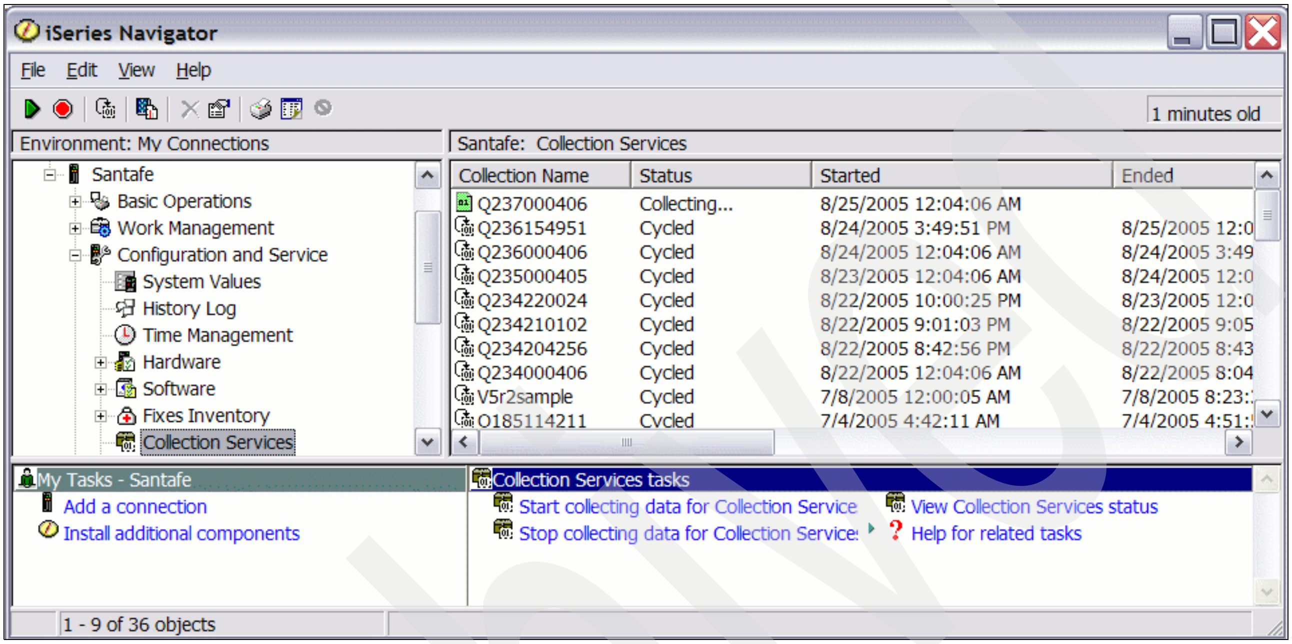


```
                            Collect Performance Data                   SYSNAME
                                                               07/25/05  13:09:24
Collection Services status:
  Status  . . . . . . . . . . . . :   Started
  Collection object . . . . . . . :   Q206130025
    Library . . . . . . . . . . . :   QMPGDATA
  Started . . . . . . . . . . . . :   07/25/05  13:00:25
  Default collection interval . . :   00:05:00
  Retention period  . . . . . . . :   30 days 00 hours
  Cycle time  . . . . . . . . . . :   00:00:00
  Cycle interval  . . . . . . . . :   24
  Collection profile  . . . . . . :   *STANDARDP

Select one of the following:

     1. Start Performance Collection
     2. Configure Performance Collection
     3. End Performance Collection

Selection or command
===>

F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel
```

Figure 3-14 Option 2 (Collect performance data) from the PERFORM menu

- iSeries Navigator

  iSeries Navigator provides a graphical user interface (GUI) for the configuration and management of Collection Services. See Figure 3-15. This is the most comprehensive interface to Collection Services. You can view and change the configuration properties, start and stop the collection, and work with the management collection objects that exist on your system. You can even create customized collection profiles.


iSeries Navigator
File
Edit
View
Help
1 minutes old
Environment: My Connections
Santafe
Basic Operations
Work Management
Configuration and Service
System Values
History Log
Time Management
Hardware
Software
Fixes Inventory
Collection Services
Santafe: Collection Services
My Tasks - Santafe
Add a connection
Install additional components
Collection Services tasks
Start collecting data for Collection Service
Stop collecting data for Collection Service
View Collection Services status
Help for related tasks
1 - 9 of 36 objects

Collection Name	Status	Started	Ended
Q237000406	Collecting...	8/25/2005 12:04:06 AM	
Q236154951	Cycled	8/24/2005 3:49:51 PM	8/25/2005 12:0
Q236000406	Cycled	8/24/2005 12:04:06 AM	8/24/2005 3:49
Q235000405	Cycled	8/23/2005 12:04:06 AM	8/24/2005 12:0
Q234220024	Cycled	8/22/2005 10:00:25 PM	8/23/2005 12:0
Q234210102	Cycled	8/22/2005 9:01:03 PM	8/22/2005 9:05
Q234204256	Cycled	8/22/2005 8:42:56 PM	8/22/2005 8:43
Q234000406	Cycled	8/22/2005 12:04:06 AM	8/22/2005 8:04
V5r2sample	Cycled	7/8/2005 12:00:05 AM	7/8/2005 8:23:
Q185114211	Cycled	7/4/2005 4:42:11 AM	7/4/2005 4:51:

Figure 3-15 iSeries Navigator GUI for the configuration and management of Collection Services

3.10.3 Which configuration is recommended

The defaults settings are sufficient for most users. You should have Collection Services running the profile collection as a matter of standard system configuration.

Normally PM iSeries has set minimum collection criteria for the system and ensures Collection Services is running. As of V5R2, this is true even if you are not using PM iSeries to send data to IBM. PM iSeries sets up the *STANDARDP profile (collects all system and communications protocol data) at 15 minute intervals, and CRTPFRDTA is run concurrently. The QMPGDATA library is used for the collection objects and files.

If you need to start Collection Services on your own, use one of the start methods described earlier. Collection Services then runs automatically until your system is shut down or you manually stop it. You can use either the command or API in your system startup procedure.

You should consider the following most important configuration parameters.

- The collection interval

  Fifteen minutes is the maximum recommended. Five minutes is better if you have no system storage or utilization concerns. If you are doing detailed analysis, consider collecting more often based on your needs.

- Retention period

  This is how long the management collection objects remain on your system before Collection Services automatically deletes them. The default is one day, but a longer duration is better if you have storage available; we recommend one week. This way if a problem occurs, the data is available should you want to export additional data.

> **Note:** iSeries Navigator, 5250 Config/Start Collector, and PM iSeries all have interfaces for how long to save the collected data. The shortest time period is the one used to determine how long the collected data remains on your system.

- Whether you want to create database files during collection

  If you are not planning to use the data on a regular basis and have PM iSeries disabled, you do not need to do this. You can always do it later if you saved the management collection object.

Refer to the iSeries Information Center to learn more.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzahx/rzahxcollectdatacs.htm

3.11 Performance Management iSeries (PM iSeries)

The PM iSeries tool is a performance tool that helps you collect data continuously, and easily, without having a thorough understanding of the performance data. PM iSeries automates the running of Collection Services and presents this data in the form of graphs or bar charts. It can also be used in conjunction with Collection Services to collect data for other purposes, such as response time per interactive user or disk used for each signed on user.

You can also use PM iSeries to gain an overall idea of how your system is running and to follow trends of your system. For instance, it is good to have a set of data to act as a baseline reference before you have performance problems. It is helpful to understand how your system was running normally and when and how it changed. You can also use this reference data when you upgrade your system or add or upgrade applications by allowing you to make before and after comparisons.

PM iSeries is intended to be a high-level tool. You can let it run continuously and then you can run other performance tools on the data created with it. For example, PM iSeries automatically starts Collection Services. It generates the QAPM* database files and reduces them into a summarized format for transmission to IBM. By default, 60 days of this summarized data is kept on your system.

If you want, PM iSeries can transmit the data back to IBM. The real benefit is gained then. IBM creates high level performance reports for you to view. A 15-month history is maintained, and this data is available for upgrade sizing by using the Workload Estimator (WLE). With the first transmission, a user ID and password are sent to you to access these reports. You can find a sample of the reports at the PM eServer iSeries Report Examples site.

http://www-03.ibm.com/servers/eserver/iseries/pm/guided_tour/examples.html

PM iSeries is one of the high level tools that you should run continuously. Other functions provided by PM iSeries include the generation of various reports to assist in the Capacity Planning and Performance Analysis of the system.

The PM iSeries reports can provide the information needed to:

- Forecast data processing growths based on trends
- Plan and maximize return on hardware investments
- Identify performance bottlenecks
- Size the next system upgrade through integration with the Work Load Estimator

Using the reports generated from the data collected and combining them with trend analysis, you can forecast data processing growths. This analysis can be based on up to 15 months of data collected from the system. You can use the PM iSeries graphs to plan when additional investment is required for new hardware.

When you experience performance problems or IT processing bottlenecks, you can use the PM iSeries interactive interface to focus the reports on a particular day and determine the time of day for the performance problem or when the bottleneck occurred. Using this information, you can determine what needs to be done to fix the problem. This may involve ordering additional disk, memory, or a larger processor.

For more information about PM iSeries, go to the PM eServer iSeries Web site.

http://www.ibm.com/eserver/iseries/pm

Also, refer to the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm?info/rzahx/rzahxplangrow1pm1.htm

3.12 IBM eServer Workload Estimator

The purpose of the IBM @server Workload Estimator is to provide sizing recommendations for the IBM brand of servers running one or more workloads associated with On Demand Business, collaboration, or both, such as Domino, Web Serving, or WebSphere. You can also use WLE to size new systems with all new workloads. Or you can use it to size the upgrade of an existing system running the original workload or running a combination of new workloads along with the original workload.

WLE recommends the model, memory, and disk requirements that are necessary to meet reasonable performance expectations. In the case of an iSeries system, WLE also recommends the 5250 OLTP feature. To use WLE, you select one or more workloads and answer a few questions about each workload. Based on the answers, WLE generates a recommendation and shows the predicted CPU utilization of the recommended system in graphical format. You can print the final results or generate them in the Portable Document Format (PDF). Advanced users have the option to provide more specific information for particular workloads, to gain a more accurate estimate.

Sizing recommendations start with benchmarks and performance measurements based on well-defined, consistent workloads. We have done those measurements and benchmarks for numerous IBM @server systems. We also develop rules of thumb for relating these performance measurements to other workloads that don't match the typical measured workload. We've used our experience running a large number of users in a closed benchmark environment, along with feedback from our clients and Business Partners who have ported their applications, to develop these rules of thumb. However, keep in mind that many of these technologies are constantly evolving. We will continue to refine these sizing recommendations as IBM and our clients gain more experience.

As with every performance estimate, whether a rule of thumb or a sophisticated model, you always need to treat it as an estimate. This is particularly true with robust IBM @server systems that offer so many different capabilities, where each installation has unique performance characteristics and demands. The typical disclaimers that go with any performance estimate ("your experience might vary...") are especially true. We provide these sizing estimates as general guidelines, but can't guarantee their accuracy in all circumstances.

3.12.1 How to use WLE

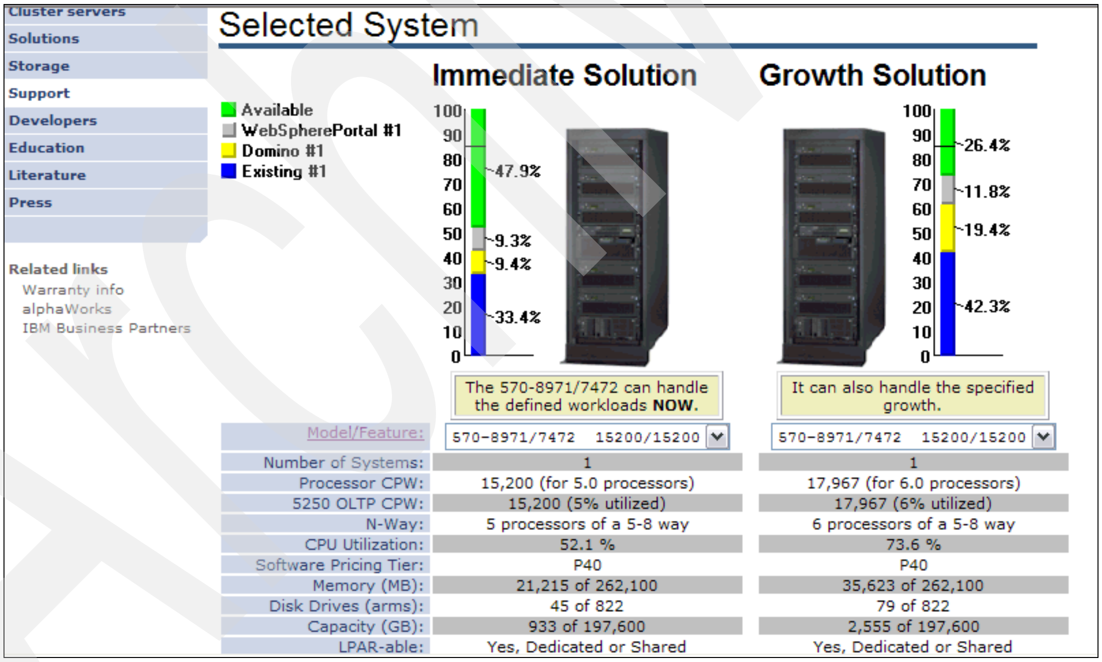
You start WLE by pointing your Web browser, using Microsoft® Internet Explorer or Netscape, to the following address:

http://www.ibm.com/eserver/iseries/support/estimator

Then follow these steps to use WLE:

1. If the system to be sized contains multiple partitions, add the partitions to WLE by clicking the **Add Partition** link.
2. For each partition, add the workloads that will run in the partition by clicking the **Add Workload** link.
3. Define the characteristics of each workload by answering questions about such information as the number of users for the application and the number of Web pages served.

After the characteristics of each workload are defined, WLE recommends two systems. The first system is able to run the workloads as currently defined by the user of WLE. The second system is able to run the workloads even if they grow at a default rate of 50%. You can adjust the growth rate to reflect the expected growth rate for this system. Figure 3-16 shows an example of the results produced for a sizing that contains three workloads, an existing model 840, a new Domino workload, and a new WebSphere Portal workload. The recommended system is an IBM POWER5™ system @server i5 model 570.



Cluster servers
Solutions
Storage
Support
Developers
Education
Literature
Press

Related links
Warranty info
alphaWorks
IBM Business Partners

Selected System

Immediate Solution

Growth Solution

The 570-8971/7472 can handle the defined workloads **NOW.**

It can also handle the specified growth.

Model/Feature:	570-8971/7472 15200/15200	570-8971/7472 15200/15200
Number of Systems:	1	1
Processor CPW:	15,200 (for 5.0 processors)	17,967 (for 6.0 processors)
5250 OLTP CPW:	15,200 (5% utilized)	17,967 (6% utilized)
N-Way:	5 processors of a 5-8 way	6 processors of a 5-8 way
CPU Utilization:	52.1 %	73.6 %
Software Pricing Tier:	P40	P40
Memory (MB):	21,215 of 262,100	35,623 of 262,100
Disk Drives (arms):	45 of 822	79 of 822
Capacity (GB):	933 of 197,600	2,555 of 197,600
LPAR-able:	Yes, Dedicated or Shared	Yes, Dedicated or Shared

Figure 3-16 WLE recommendations provided

As shown in Figure 3-16, the recommendations provided by WLE include the processor model, the amount of total commercial processing workload (CPW), the 5250 online transaction processing (OLTP) CPW required to run the workloads, and the software tier required for the recommended model. Other key resource recommendations include the estimated CPU utilization, the amount of memory, disk arms, and disk storage.

There are many additional features in WLE that are described in several tutorials provided with the tool. Click the **Help/Tutorial** link to access the basic and detailed tutorials.

3.13 Performance Adjuster

Performance Adjuster, a built-in function in i5/OS, has the ability to automatically manage the shared storage pools without user interaction. The Performance Adjuster function is controlled by the QPFRADJ system value. The operating system is shipped with this function enabled by default, but you can easily disable it by setting the QPFRADJ system value to zero, which means “no adjustment”. There are other values, such as Dynamic Priority Scheduling, Dynamic Priority Adjust, and Expert Cache, that allow controlling of system performance.

Performance Adjuster runs in a system job, QPFRADJ, and monitors basic performance attributes of the shared storage pools. Adjuster mainly looks at two things: activity level and number of page faults occurring in each shared storage pool. Adjuster keeps the activity levels of all the pools high enough to avoid CPU starvation of active threads, but not too high so that system resources are wasted on maintaining unused control structures.

Working with page faults is more complicated. Performance Adjuster monitors page faults and pool sizes for all shared storage pools. It tries to move storage between the pools so that to keep overall page fault rate of entire system to a minimum.

Performance Adjuster is highly tunable. You can use the WRKSHRPOOL system command to change various storage pool settings used by the Performance Adjuster. You can also use the WRKSHRPOOL command to specify pool size limits, relative priorities, and page fault rate guidelines on the individual storage pool level. You can direct jobs that belong to different types of applications to different subsystems or pools and carefully choose parameters for the pools. In doing so, you can implement fairly sophisticated pool management strategies. This is also discussed in 3.14, “Work with Shared Pools (WRKSHRPOOL) command” on page 42.

Be aware of the following items when using Performance Adjuster.

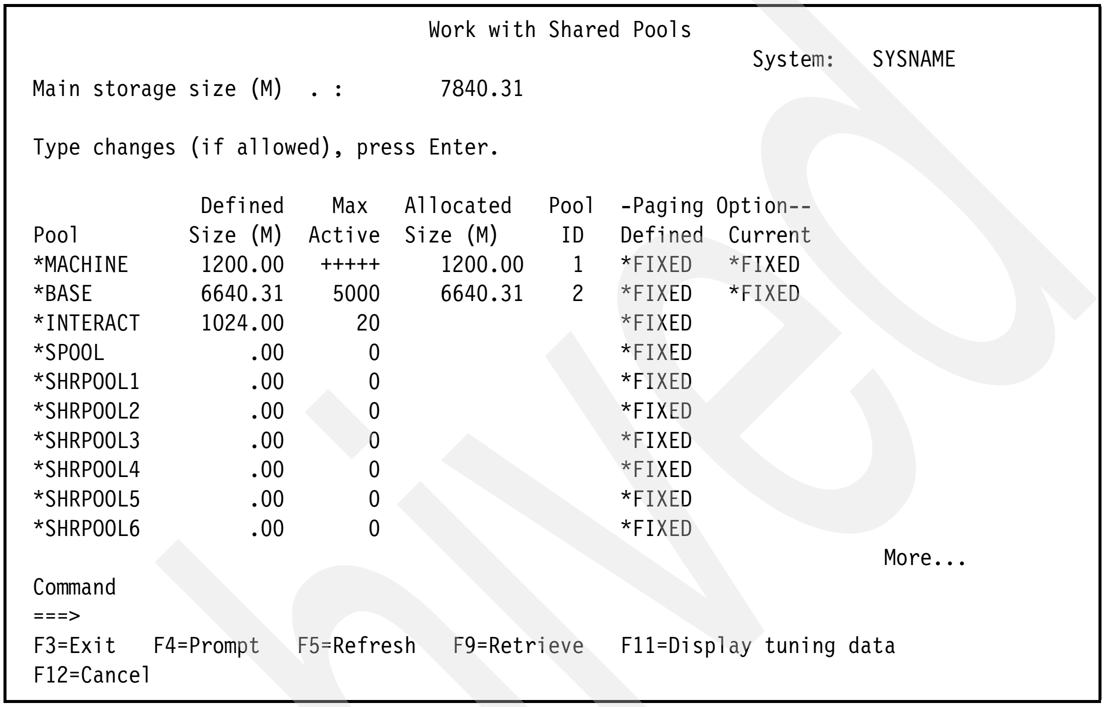
- Performance Adjuster only works with shared storage pools; subsystem private pools are never touched. This allows isolating certain types of work from the influence of the Performance Adjuster.
- Adjuster relies on periodic statistical sampling of the storage pool behavior. Therefore, if you just made a change in your application usage, you need to wait for a reasonable period of time, for example 15 minutes, for the Adjuster to react to it. Do not expect the adjuster to immediately react to short-term changes in the application behavior.
- Adjuster makes changes to the pool configuration in increments. When the application usage pattern changes significantly, such as a change between night and day shifts, it takes time for the adjuster to recognize the change and tune the storage pool configuration. During this time, the system may appear sluggish. When such significant change is anticipated, especially if it is part of a regular schedule, it might be more efficient to make explicit changes to the pool configuration.

To learn more about how Performance Adjuster works and how to use it efficiently, see the Performance Adjuster experience report in the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/experience/adjuster.pdf

3.14 Work with Shared Pools (WRKSHRPOOL) command

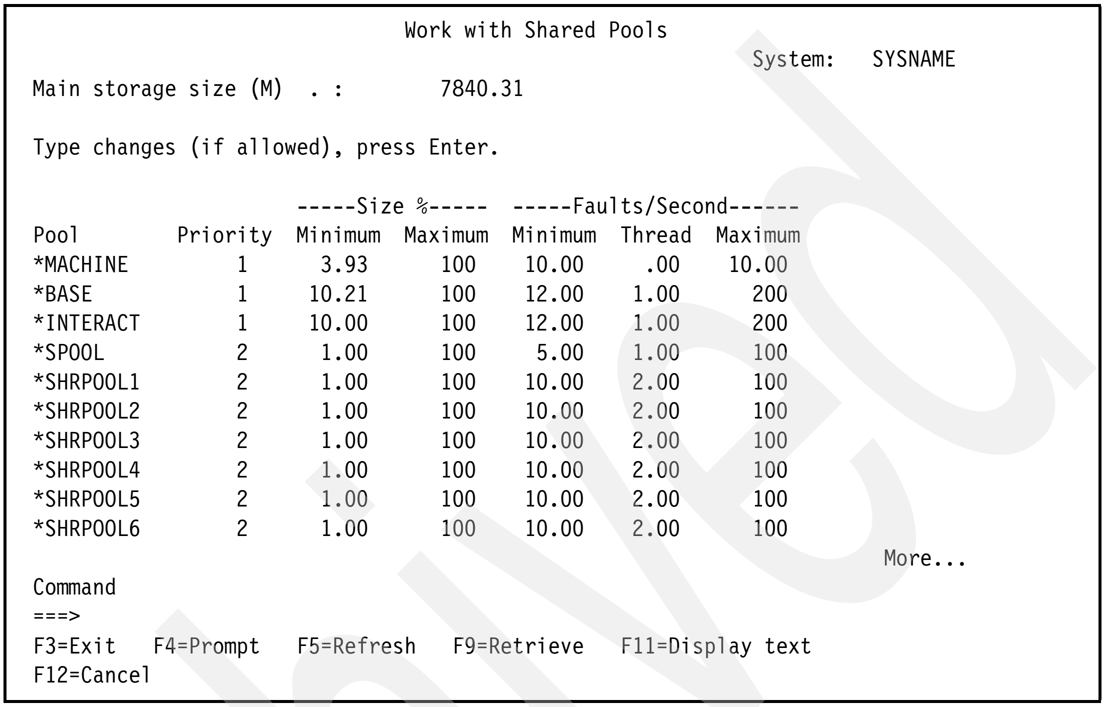
The WRKSHRPOOL command is part of i5/OS. It displays the names and status of the shared storage pools. You can use this command to change the size and maximum activity level for each pool. Refer to 3.13, “Performance Adjuster” on page 41, to learn more about this tool. Figure 3-17 shows a sample of the output of this command.



```
                                  Work with Shared Pools
                                                                System:   SYSNAME
 Main storage size (M)  . :        7840.31

 Type changes (if allowed), press Enter.

               Defined    Max   Allocated   Pool  -Paging Option--
 Pool          Size (M)  Active  Size (M)    ID   Defined  Current
 *MACHINE      1200.00   +++++     1200.00    1   *FIXED   *FIXED
 *BASE         6640.31    5000     6640.31    2   *FIXED   *FIXED
 *INTERACT     1024.00      20                    *FIXED
 *SPOOL            .00       0                    *FIXED
 *SHRPOOL1         .00       0                    *FIXED
 *SHRPOOL2         .00       0                    *FIXED
 *SHRPOOL3         .00       0                    *FIXED
 *SHRPOOL4         .00       0                    *FIXED
 *SHRPOOL5         .00       0                    *FIXED
 *SHRPOOL6         .00       0                    *FIXED
                                                                           More...
 Command
 ===>
 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display tuning data
 F12=Cancel
```

Figure 3-17 WRKSHRPOOL sample output

If you press the F11 key, you see details about the size of each pool (percent of total memory) and the faulting. In addition, you can view and change faulting values to be used by the Performance Adjuster if the QPFRADJ system value is set to 2 or 3 (automatic performance adjust). Figure 3-18 shows a sample of the output.



```
                                 Work with Shared Pools
                                                                System:   SYSNAME
 Main storage size (M)  . :        7840.31

 Type changes (if allowed), press Enter.

                      -----Size %-----  -----Faults/Second------
 Pool        Priority  Minimum  Maximum  Minimum  Thread  Maximum
 *MACHINE        1       3.93      100    10.00      .00    10.00
 *BASE           1      10.21      100    12.00     1.00      200
 *INTERACT       1      10.00      100    12.00     1.00      200
 *SPOOL          2       1.00      100     5.00     1.00      100
 *SHRPOOL1       2       1.00      100    10.00     2.00      100
 *SHRPOOL2       2       1.00      100    10.00     2.00      100
 *SHRPOOL3       2       1.00      100    10.00     2.00      100
 *SHRPOOL4       2       1.00      100    10.00     2.00      100
 *SHRPOOL5       2       1.00      100    10.00     2.00      100
 *SHRPOOL6       2       1.00      100    10.00     2.00      100
                                                                          More...
 Command
 ===>
 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display text
 F12=Cancel
```

Figure 3-18 WRKSHRPOOL: Using F11 to display the tuning data output

3.15 Start Trace (STRTRC) and End Trace (ENDTRC) commands

The STRTRC and ENDTRC commands are part of i5/OS. They are used to start and stop job-level traces of OPM and ILE programs.

When you run the STRTRC command, you specify the type of trace with the JOBTRCTYPE keyword:

- *FLOW (flow of control: call or return)
- *DATA (predefined trace points)
- *TRCTYPE (one or more components specified with the TRCTYPE keyword)

You can specify one or more jobs to trace or *ALL.

The ENDTRC command ends the trace and direct the output trace records to a specific library. It can also print a report.

3.16 Performance Tools for iSeries LPP (57XX-PT1)

Performance Tools for iSeries is a set of tools for viewing, analyzing, reporting, and graphing performance data collected by Collection Services. This includes:

- Reports
- Commands
- Graphs
- Display Performance Data GUI (Performance Tools plug-in for iSeries Navigator)

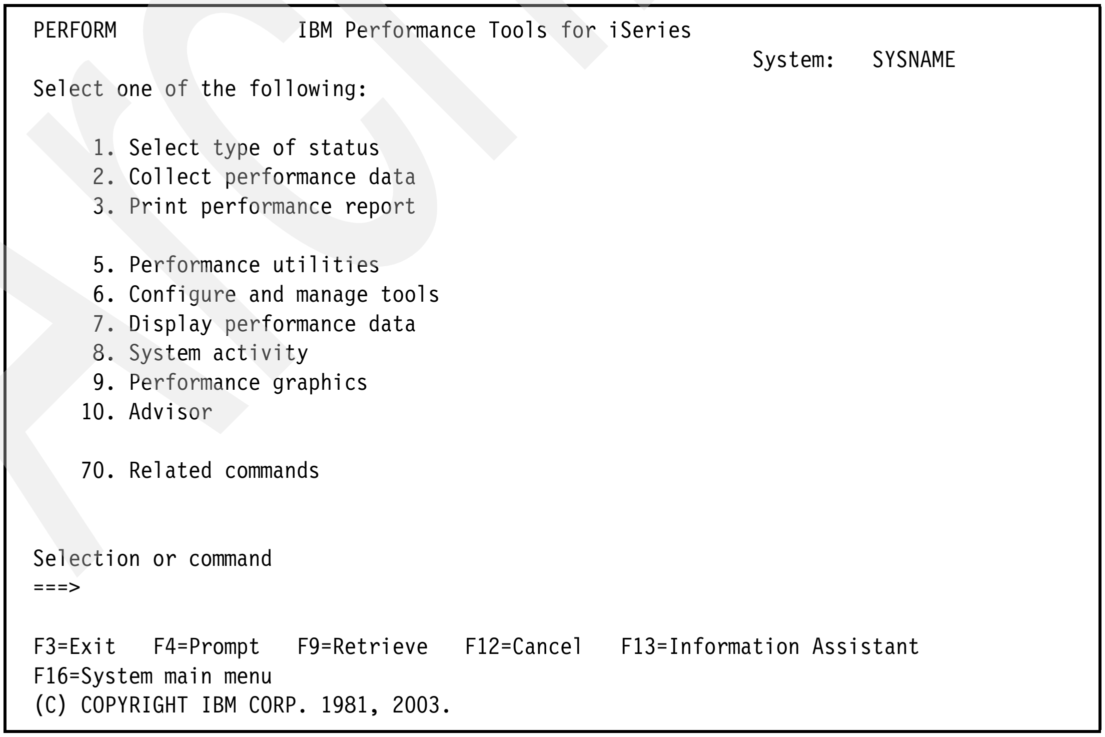
Usually, you collect performance data for trend analysis and performance analysis, but raw Collection Services performance data is difficult to understand. To facilitate this work, Performance Tools for iSeries offers tools to process and organize performance data. You can print a variety of reports that organize performance data into a logical and useful format (see 3.16.2, “Performance reports” on page 45).

In the same way, you can view performance data, display graphs to show trends, and analyze your system performance. For this purpose, you can use interactive commands such as Display Performance Data (DSPPFRDTA) and Analyze Performance Data (ANZPFRDTA). Or you can display performance data from the Display Performance Data GUI.

Using the Performance Tools LPP is often the first step to investigate performance problems on your system.

3.16.1 How to use Performance Tools

The Performance Tools LPP offers you a menu interface that provides access to all of the performance tools. You can use the STRPFRT command or the GO PERFORM command on any command line to call the Performance Tools main menu interface. Figure 3-19 shows the Performance Tools menu interface.



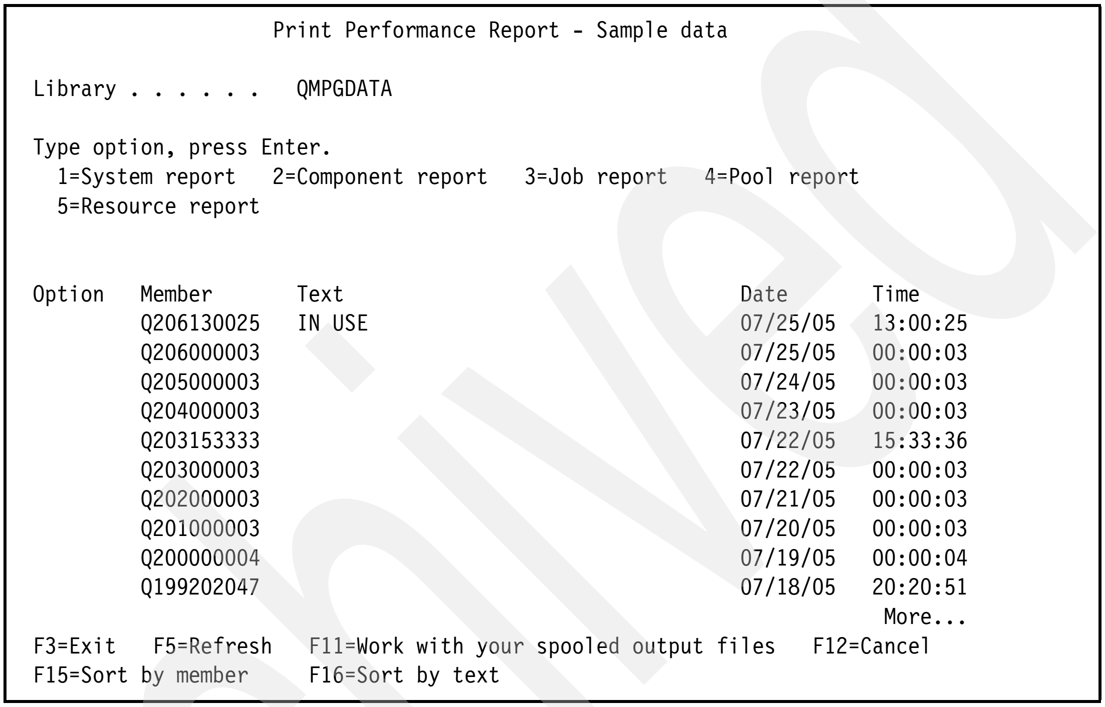
```
PERFORM               IBM Performance Tools for iSeries
                                                             System:   SYSNAME
Select one of the following:

     1. Select type of status
     2. Collect performance data
     3. Print performance report

     5. Performance utilities
     6. Configure and manage tools
     7. Display performance data
     8. System activity
     9. Performance graphics
    10. Advisor

    70. Related commands


Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F16=System main menu
(C) COPYRIGHT IBM CORP. 1981, 2003.
```

Figure 3-19 PERFORM menu

On this menu, you find options to print performance reports, display performance data, and analyze performance data.

3.16.2 Performance reports

To run a performance report, you select option 3 from the Performance Tools menu. Then you see a variety of performance reports to help you isolate performance problems on your system.



```
                      Print Performance Report - Sample data

 Library . . . . . .   QMPGDATA

 Type option, press Enter.
   1=System report   2=Component report   3=Job report   4=Pool report
   5=Resource report


 Option   Member       Text                                 Date       Time
          Q206130025   IN USE                               07/25/05   13:00:25
          Q206000003                                        07/25/05   00:00:03
          Q205000003                                        07/24/05   00:00:03
          Q204000003                                        07/23/05   00:00:03
          Q203153333                                        07/22/05   15:33:36
          Q203000003                                        07/22/05   00:00:03
          Q202000003                                        07/21/05   00:00:03
          Q201000003                                        07/20/05   00:00:03
          Q200000004                                        07/19/05   00:00:04
          Q199202047                                        07/18/05   20:20:51
                                                                        More...
 F3=Exit   F5=Refresh   F11=Work with your spooled output files   F12=Cancel
 F15=Sort by member     F16=Sort by text
```

Figure 3-20 PERFORM menu: Option 3 to print the performance report

Based on the collected performance data, the performance reports can show you how system resources are being used. You can also find the jobs, system tasks, users, or application programs that might be causing performance problems.

However, sometimes it is not easy to know which is the best report for your performance analysis. Table 3-1 shows a brief overview of the performance reports and explains when and why you should use each one of them.

Table 3-1 Overview of performance reports

Report	Description	What is shown	How you use the information
System	This report gives an overview of how the system is operating over a selected time. Print this report often to give you and idea of your system use. Use the PRTSYSRPT command to create this report.	System workload, resource utilization, storage pool utilization, disk utilization, Communication summary, TCP/IP summary, and HTTP server summary	Workload projection
Component	This report expands on the information given in the System report. It shows detail for each interval to help you find which jobs are consuming high amounts of system resources, such as CPU and disk. Use the PRTCPTRPT command to create this report.	Interval activity, job workload activity, storage pool activity, disk activity, IOP utilizations, local workstation, exception occurrence, database journaling, TCP/IP activity, HTTP server activity	Job analysis, resource analysis, and hardware growth
Job Interval	This job-oriented report is organized by time interval. It helps you find, by interval, which jobs are consuming high amounts of system resources, such as CPU and disk. Use the PRTJOBRPT command to create this report.	Interactive job summary, noninteractive job summary, interactive job detail, and noninteractive job detail	Job analysis
Pool	The pool-oriented report is organized by time interval. It helps you find which subsystems or pools are consuming high amounts of system resources, such as CPU and disk. Use the PRTPOLRPT command to create this report.	Subsystem activity, and pool activity	Pool analysis
Resource	The device resource usage report is organized by time interval. This report shows how system resources are used. Use the PRTRSCRPT command to create this report.	Disk utilization, communications line detail, IOP utilizations, local work station, and remote work station	System resource use
Transaction	This report is based on performance trace data that is collected by STRPFRTRC or ENDPFRTRC. Use the PRTTNSRPT command to create this report.	Summary level, transaction level, and transition level reports are available	Job analysis
Lock	This report locks and seizes information in Performance Trace Data. Use the PRTLCKRPT command to create this report.	The report options display information that is sorted by holder, requestor, time of day, and object	Resource contention analysis.
Activity	The report data is collected by WRKSYSACT OUTPUT(*FILE) or WRKSYSACT OUTPUT(*BOTH). Use the PRTACTRPT command to create this report.	Sections in the report are sorted by CPU utilization, total I/O, and wait time	Job analysis

3.16.3 Display Performance Data (DSPPFRDTA) command

The DSPPFRDTA is an interactive command. It summarizes performance data that is collected with Collection Services and provides detailed information about jobs, subsystems, pool, disk, and communications. This tool helps you to isolate performance problems to a certain area.

3.16.4 Analyze Performance Data (ANZPFRDTA) command

The ANZPFRDTA command, also known as the *Advisor*, analyzes performance data, makes suggestions, and offers recommendations for improving performance. The Advisor examines most of the performance database files to find performance problems on the system. It is a helpful tool if you do not have enough experience in analyzing performance problems with performance reports; all recommendations and conclusions are easy to understand. You will find useful information, such as total disk I/O operations, top ten lists of jobs, and the Highest Interactive Feature Utilization interval, in the Conclusion section of the Advisor.

Use the tune system key (F9) to adjust your system according to the recommendations.

> **Note:** To use the Tune System function, you must run the Advisor on the same system on which the data was collected.

Refer to the following Web sites for more information.

- iSeries Information Center
  - Performance Tools LPP

    `http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm?info/rzahx/rzahxperftoolsdesc.htm`
  - Performance Data GUI

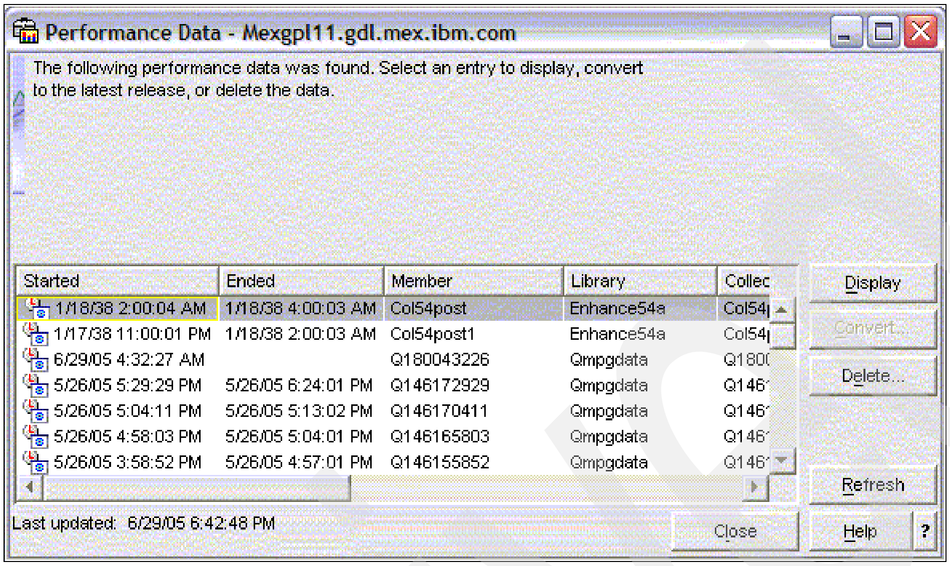
    `http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm?info/rzahx/rzahxperftoolcondisplay.htm`
  - Performance Tools reports

    `http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzahx/rzahxreportperftools.htm`
- *Performance Tools for iSeries*, SC41-5340-01
  - Display Performance Data
  - Analyze Performance Data

3.17 Performance Tools plug-in (Display Performance Data GUI)

As mentioned earlier, the Display Performance Data GUI is a plug-in for iSeries Navigator. It is part of the Performance Tools LPP (not shipped with the operating system) and is based on Collection Services performance data. You can use it to view performance data, display graphs, and produce the performance reports (System report, Component report, Resource report, etc.) from the iSeries Navigator.

The first window (Figure 3-21) shows all the collection objects (MGTCOL) in the system and buttons to display, convert or delete the performance data based on those collection objects. The Convert and Delete functions are similar to the CVTPFRDTA and DLTPFRDTA commands respectively.



The following performance data was found. Select an entry to display, convert to the latest release, or delete the data.

Started	Ended	Member	Library	Collec
1/18/38 2:00:04 AM	1/18/38 4:00:03 AM	Col54post	Enhance54a	Col54
1/17/38 11:00:01 PM	1/18/38 2:00:03 AM	Col54post1	Enhance54a	Col54
6/29/05 4:32:27 AM		Q180043226	Qmpgdata	Q1800
5/26/05 5:29:29 PM	5/26/05 6:24:01 PM	Q146172929	Qmpgdata	Q1461
5/26/05 5:04:11 PM	5/26/05 5:13:02 PM	Q146170411	Qmpgdata	Q1461
5/26/05 4:58:03 PM	5/26/05 5:04:01 PM	Q146165803	Qmpgdata	Q1461
5/26/05 3:58:52 PM	5/26/05 4:57:01 PM	Q146155852	Qmpgdata	Q1461

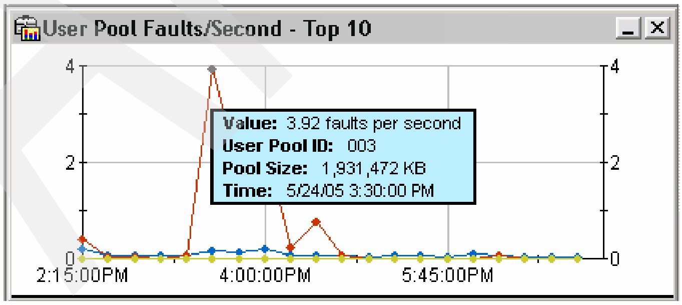
Last updated: 6/29/05 6:42:48 PM

Figure 3-21 Display Performance Data GUI: First window showing all collection objects on the system

In the Graphs pane, you can see graphs for the following metrics.

- Transaction Count
- Transaction Response Time
- Total CPU Utilization
- Interactive CPU utilization
- Batch CPU Utilization
- Interactive Feature Utilization
- High Disk Utilization
- Machine Pool Faults/Second
- User Pool Faults/Second (A Top 10 pools graph)
- Exception CPU Utilization

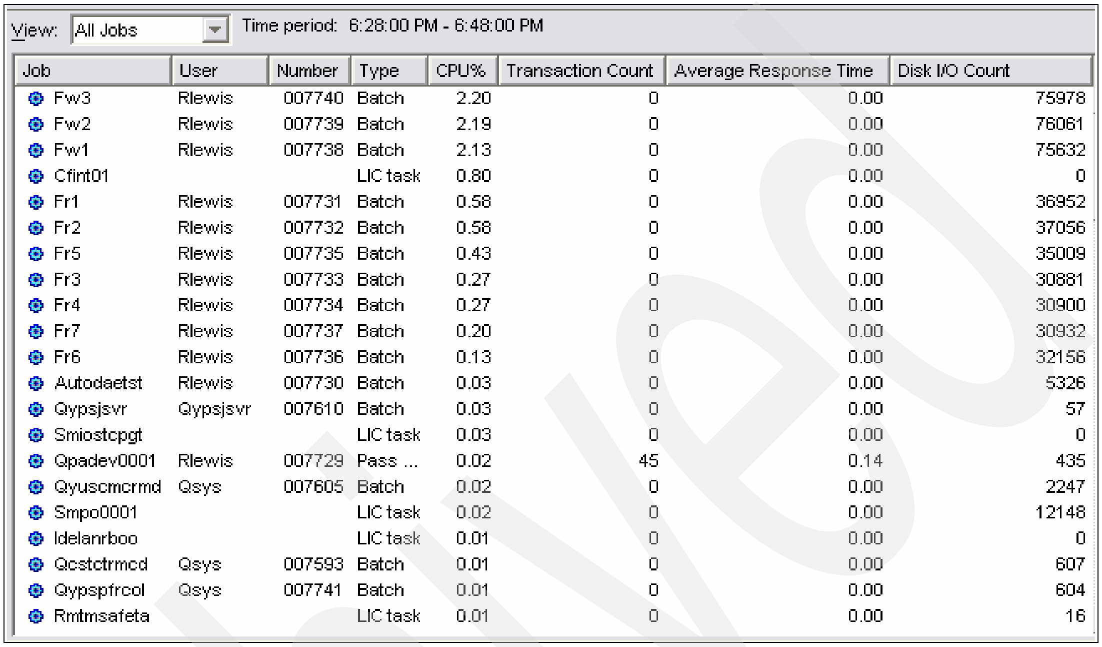
Figure 3-22 shows an example of one of the graphs.



Value: 3.92 faults per second
User Pool ID: 003
Pool Size: 1,931,472 KB
Time: 5/24/05 3:30:00 PM

Figure 3-22 User Pool Faults/Second graph

The Details pane (Figure 3-23) allows you to view detailed performance data in a variety of ways. For example, you can view all jobs and tasks running during the performance data collection and performance statistics arranged by Subsystem, Job type, Time Interval, Memory Pool or Disk Unit. This function is similar to the 5250-based DSPPFRDTA command. See 3.16, “Performance Tools for iSeries LPP (57XX-PT1)” on page 44, for more information.



View: All Jobs Time period: 6:28:00 PM - 6:48:00 PM

Job	User	Number	Type	CPU%	Transaction Count	Average Response Time	Disk I/O Count
Fw3	Rlewis	007740	Batch	2.20	0	0.00	75978
Fw2	Rlewis	007739	Batch	2.19	0	0.00	76061
Fw1	Rlewis	007738	Batch	2.13	0	0.00	75632
Cfint01			LIC task	0.80	0	0.00	0
Fr1	Rlewis	007731	Batch	0.58	0	0.00	36952
Fr2	Rlewis	007732	Batch	0.58	0	0.00	37056
Fr5	Rlewis	007735	Batch	0.43	0	0.00	35009
Fr3	Rlewis	007733	Batch	0.27	0	0.00	30881
Fr4	Rlewis	007734	Batch	0.27	0	0.00	30900
Fr7	Rlewis	007737	Batch	0.20	0	0.00	30932
Fr6	Rlewis	007736	Batch	0.13	0	0.00	32156
Autodaetst	Rlewis	007730	Batch	0.03	0	0.00	5326
Qypsjsvr	Qypsjsvr	007610	Batch	0.03	0	0.00	57
Smiostcpgt			LIC task	0.03	0	0.00	0
Qpadev0001	Rlewis	007729	Pass ...	0.02	45	0.14	435
Qyuscmcrmd	Qsys	007605	Batch	0.02	0	0.00	2247
Smpo0001			LIC task	0.02	0	0.00	12148
Idelanrboo			LIC task	0.01	0	0.00	0
Qcstctrmcd	Qsys	007593	Batch	0.01	0	0.00	607
Qypspfrcol	Qsys	007741	Batch	0.01	0	0.00	604
Rmtmsafeta			LIC task	0.01	0	0.00	16

Figure 3-23 Details pane of Display Performance Data

From the Reports menu, you can submit a System, Component, Job, Pool or Resource report. After the report is processed on the iSeries, you can view the associated spooled file with an Advanced Function Presentation™ (AFP™) viewer. This graphical view of the performance data enables quick detection of abnormally high or low resource utilizations. It also identifies the jobs and tasks that are active at that time to make it easier for you to manage performance of your iSeries server.

3.17.1 How to use the Performance Tools plug-in

Sign on to a system shown under My Connections, and display a list of performance database file member names. From this list, you select a member to display using the Display Performance Data Graphics function.

- Option 1

  a. Expand the **Configuration and Services** folder.

  b. Double-click the **Collection Services** folder.

  c. This opens a list of all management collection objects that are currently on the system. It includes all cycled status collection objects. If Collection Services is active, it includes the management collection object associated with the active collection. Right-click any of these objects and select **Performance Tools** → **Performance Data**.

  d. The Performance Data window opens. It shows a list of all performance database member names created from that collection object.

- Option 2

  a. Expand the **Configuration and Services** folder.

  b. Right-click **Collection Services** and select **Performance Tools** → **Performance Data**.

  c. The Performance Data window opens that lists all performance database member names stored on the entire system. Scroll to the right to see all the columns of information for each performance database member. This window also shows (in the lower subwindow as indicated by the arrow) the columns of information to the right.

For more details about Performance Data GUI, refer to the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm?info/rzahx/rzahxperftoolcondisplay.htm

3.18 iDoctor for iSeries Job Watcher

The iDoctor for iSeries Job Watcher is a performance tool that returns real-time information about a selected set of jobs, threads, or Licensed Internal Code (LIC) tasks. It is similar in sampling function to the WRKACTJOB and WRKSYSACT system commands in which each refresh computes delta information. In Job Watcher, these refreshes can be set to occur automatically, as frequently as every 50 milliseconds. Furthermore, Job Watcher collects the data from the jobs, threads, or tasks being watched in a manner that does not affect other jobs on the system while it is collecting.

Job Watcher allows the user to collect a variety of different types of information about a set of jobs (or all jobs) on a system. The types of data collected include wait times, CPU, I/O activity, call stacks, SQL statements, communications, and activation group statistics. Many graphs are available to help the user interpret the data. At a high level, the graphs can illustrate what jobs are performing poorly on a system. At a low level, the user can see an interval-by-interval view of what an individual job or thread is doing.

Job Watcher provides coverage for the “middle ground” between Collection Services and Performance Explorer (PEX). In comparison to Collection Services, Job Watcher can collect different types of data at more frequent intervals and over a selected set of jobs, threads, or LIC tasks. However, Job Watcher is it is not intended to run continuously as Collection Services is. In comparison to PEX, Job Watcher provides a smaller, less complex set of data to analyze.

The data created by the tool is summarized in many different types of reports and graphs via the GUI client. The client provides a quick picture of what is happening on a per thread basis when multiple threads are being watched. The user has the flexibility to select a job or an interval and then drill down for the details while the watch is in progress or after the watch has ended. A built-in trigger mechanism allows calls to APIs or user programs based on predefined conditions so that prompt action can be taken for virtually any circumstance. These triggers are set up via the Job Watcher wizard included in the GUI client.

The information Job Watcher collects includes the following types of information among other types:

- Standard WRKSYSACT type data: CPU, disk I/O breakdown, disk space consumption, etc.
- Data currently seen only via Collection Services: Current user name, seize time, and breakdown of what types of waits (all waits) that occurred

- Data not available anywhere else in real time: Details about the current wait (duration, object, conflicting job info, specific LIC block point ID), program and procedure call stack (1000 levels deep), and more

You can find documentation for the V5R3 release of iDoctor for iSeries on the Web at:

https://www-912.ibm.com/i_dir/idoctor.nsf/F204DE4F34767E0686256F4000757A90/$FILE/iDoctorV5R3Help.pdf

3.19 iDoctor for iSeries Heap Analyzer for Java

The iDoctor for iSeries Heap Analyzer for Java provides a set of tools useful for investigating performance problems related to Java memory. The tools allow the user to perform heap analysis, advanced debug, and application review.

The first symptom of a problem related to heap growth is typically an increased rate of paging in the pool in which the Java jobs are running. This is most often caused by an uncontrollable growth of the heap size. Adding memory to the pool, treats the symptoms but does not remove the problem itself. In this type of situation, Heap Analyzer is used to perform Java application heap analysis and object create profiling (size and identification) over time. Heap Analyzer includes information about:

- Java virtual machine (JVM™) heap growth/size
- The objects being created (type of object, count and object size, object heap size)
- The application “Heap Footprint” for memory sizing and performance considerations

Heap Analyzer can run in one of two modes:

- Object table dump

  This option is similar to Dump JVM (DMPJVM). However it is a non-intrusive snapshot that provides detailed information related to object counts, object size, and actual heap size used. Heap Analyzer runs in the background and will run on enormous heaps without any affect on the JVM operation while it’s running.

- Object create profile

  This option is used to “profile” object creations in real time. To fully understand how this works, you must understand the architecture that is related to how the JVM handles object creation. Every object has two sizes associated with it: the actual object size and the heap size required by that object.

  For example, in i5/OS V5R2, there are 23 bucket sizes, and every object falls into one of these buckets. When the profiler for a JVM is turned on, it causes every object created for each bucket and thread to be logged. The logged information includes:

  - A counter for the bucket in this thread
  - The object class name
  - The block size
  - The object size
  - The object, class, and loader address
  - Stack information to identify where the object was created from

For V5R3 documentation of iDoctor for iSeries, go to the following Web site.

https://www-912.ibm.com/i_dir/idoctor.nsf/F204DE4F34767E0686256F4000757A90/$FILE/iDoctorV5R3Help.pdf

3.20 Batch Model (BCHMDL)

You can use the Batch Model (formerly known as BATCH400) tool to analyze batch runtime performance when the batch window is becoming a concern. For example, the overnight batch processing takes so long that it does not finish before the morning users sign on to the system. Or the concern may be that a specific Web application is anticipated to increase 10 times in the amount of traffic because of a new product, and it's not known if the system can handle the additional work.

Batch Model uses Collection Services data as input. The only requirement is that system, disk and job data must be collected, so *MINIMUM collections are sufficient. The collection interval time does not matter, so data collected for your other performance analysis needs can be used effectively.

Batch Model can predict batch job run times and resource usage for the following scenarios.

- **Processor changes**: Either the processor model can be changed or the number of processor units assigned to the system can be changed. In either case, the model predicts the resulting CPU utilization and CPU queuing for each batch job. You can easily tell if an upgrade will fix your batch window. If it doesn't not, then maybe the processor is not the bottleneck.
- **Disk changes**: Either the disk model/type can be changed or the number of disk drives can be changed (or both). In either case, the model predicts the resulting disk utilization and disk queuing for each batch job. Again, you can easily tell if these changes will improve your batch runtime.
- **Job/workload changes**: Many things about the job can be changed, either to try to make the job run faster or to reflect a change in the overall workload.
  - Growth rate applied

    If you anticipate that customer orders are going to increase by two times the current volume, you can apply that growth rate to the pertinent batch jobs to predict if those jobs will still run to completion in a reasonable amount of time (before all the users sign on in the morning). If the model shows the job taking too long after the growth, then you may need to add resource or alter the workload environment (see the following items).
  - Priority change

    If you have several important jobs that are not running fast enough, perhaps improving their priority will provide a sufficient increase in throughput to meet your needs. The Batch Model supports priorities from 0 through 99.
  - Concurrency change

    If you have one long running job that does not take advantage of multiple processors, you can model the effect of splitting that job up into several concurrently running batch jobs. For example, you could make four copies of the job with each copy processing one-fourth of the total amount of work. If software bottlenecks (locks and seizes) are avoided, then this can be a simple way to improve the throughput of an application without adding expensive resources.
  - Increased workload

    If you have a new application that you prototyped on a development system, Batch Model allows you to model that batch job running concurrently with other jobs on a production system. Then you will have a good idea if more resources will be needed to handle the new application when it is deployed.

- Server job changes

  Server jobs are tricky because they do not behave like traditional batch jobs. Server jobs spend much of their time waiting for work. If you have a Web server that is not very busy, but you anticipate more activity because of a promotion, you can use the Batch Model to increase how much work the server job will do per second. Server jobs don't really have a timeframe in which they need to be completed, but they need to handle a certain throughput rate, which can be accurately predicted by the Batch Model.

Refer to the *IBM @server iSeries Performance Capabilities Reference*, SC41-0607, for a short description of Batch Model. To access to this tool, consult with your IBM representative.

3.21 WebSphere tools

Refer to the WebSphere and Java section of the Performance Management Web site for papers and articles to help you take advantage of the latest performance improvements, tools, and tuning methods relating to WebSphere and Java.

http://www-03.ibm.com/servers/eserver/iseries/perfmgmt/webjtune.html

The Web site has a number of articles about WebSphere and Java performance tuning that show how to use such tools as Tivoli® Performance Viewer and PEX for particular tuning situations.

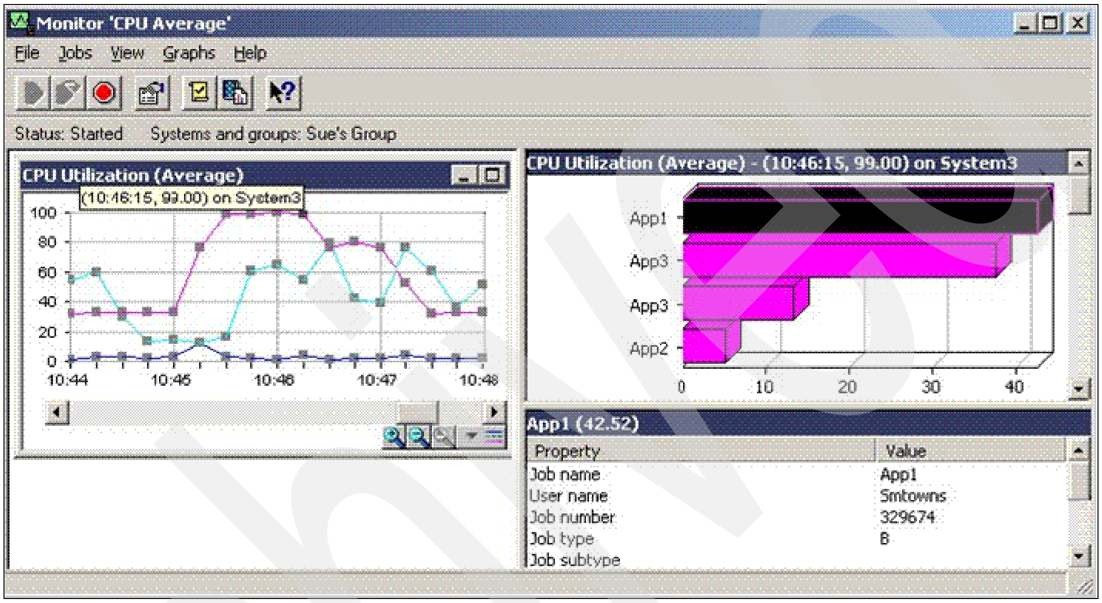
Also, refer to Chapter 4, about using performance tools for determining performance problems, in the IBM Redbook *Maximum Performance with WebSphere Application Server V5.1 on iSeries*, SG24-6383.

3.22 iSeries Navigator system monitors

Several types of monitors are available in iSeries Navigator. The type of monitor used most frequently to monitor system performance is called a *system monitor*. A system monitor gathers and presents real-time performance data that helps monitor the health of your system and identify potential performance problems before they become serious issues.

The data available in a system monitor is a subset of the information collected by Collection Services. A system monitor displays data as it is collected, for up to one hour. To view longer periods of data, use Graph History. You can change the frequency of the data collection in the monitor properties, which overrides the settings in Collection Services. You can start a system monitor on a single system or partition, or on multiple systems or partitions at the same time.

A system monitor can provide multiple levels of performance and system information. Level one in the left pane of Figure 3-24 contains the system-wide performance metrics that can be monitored such as CPU utilization, disk arm utilization, transaction rates, and interactive response time. Level two in the upper right pane in Figure 3-24 contains a list of items that contribute most to the level one metric. For example, for CPU utilization, a list of jobs is consuming the most CPU. For disk arm utilization, a list of disk arms is the busiest. Level three in the lower right pane of Figure 3-24 is a list of performance metrics and properties for the level two items. For a job consuming high amounts of CPU, it contains information such as the job priority, the pool the job is running in, the number of transactions, and database operations performed by the job.



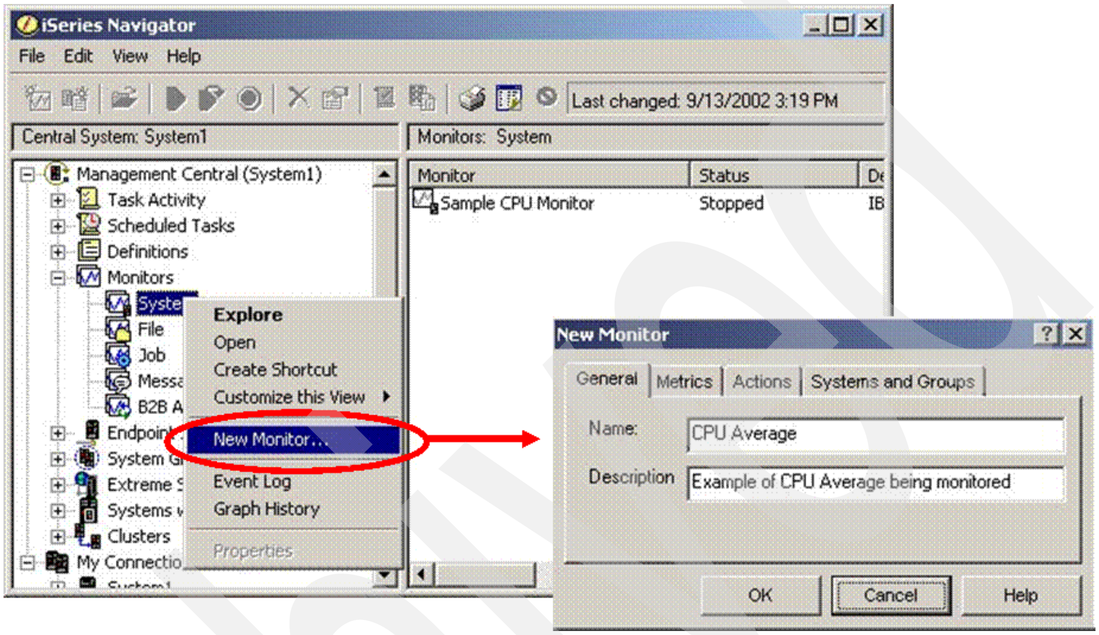
Status: Started Systems and groups: Sue's Group

Property	Value
Job name	App1
User name	Smtowns
Job number	329674
Job type	B
Job subtype	

Figure 3-24 System monitors providing multiple levels of performance and system information

One of the most valuable attributes of a System Monitor is called a *threshold*. A threshold can be defined when a monitor is created, which triggers an action when a system wide performance metric exceeds the defined “comfort level”. For example, when CPU utilization exceeds a particular level, such as 90%, a message can be sent to notify a system operator of this event. You can also call a CL command or program to perform such actions as holding a job queue or invoking paging software which notifies the operator on call that a system condition needs attention. Other actions that can be taken include the ability to log an action in an event log, open the monitor or event log on the device that is monitoring the system, or sound an alarm on the device.

System monitors can be created by right-clicking System Monitor in iSeries Navigator Management Central and selecting New Monitor as shown in Figure 3-25. When creating a monitor, you must give the monitor a name and select one or more performance metrics to be monitored. You can also define two thresholds per metric along with actions that should be taken if a threshold is reached or exceeded. And, you can define on which systems to run the monitor, or select the systems when you start the monitor.



Name: CPU Average

Description: Example of CPU Average being monitored

Figure 3-25 Creating new system monitors

When a monitor is created, you can start the monitor by right-clicking the monitor and selecting Start. For more information about how to create, start, and use System Monitors, see the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm

From the left navigation area, follow the path **Systems management** → **Performance** → **Application for performance management** → **Monitors** → **Configure monitors**.

System Monitors provide powerful capabilities to monitor what is happening on your system. When notified that a problem is occurring on your system (when a threshold is reached), a skilled system operator can often use simple systems management commands to alleviate the problem before it becomes a serious issue. However, there are times when system monitors do not help you find what caused the problem. In those situations, you need to use additional performance analysis tools, such as the Performance Tools reports and graphs, Performance Explorer, or iDoctor for iSeries.

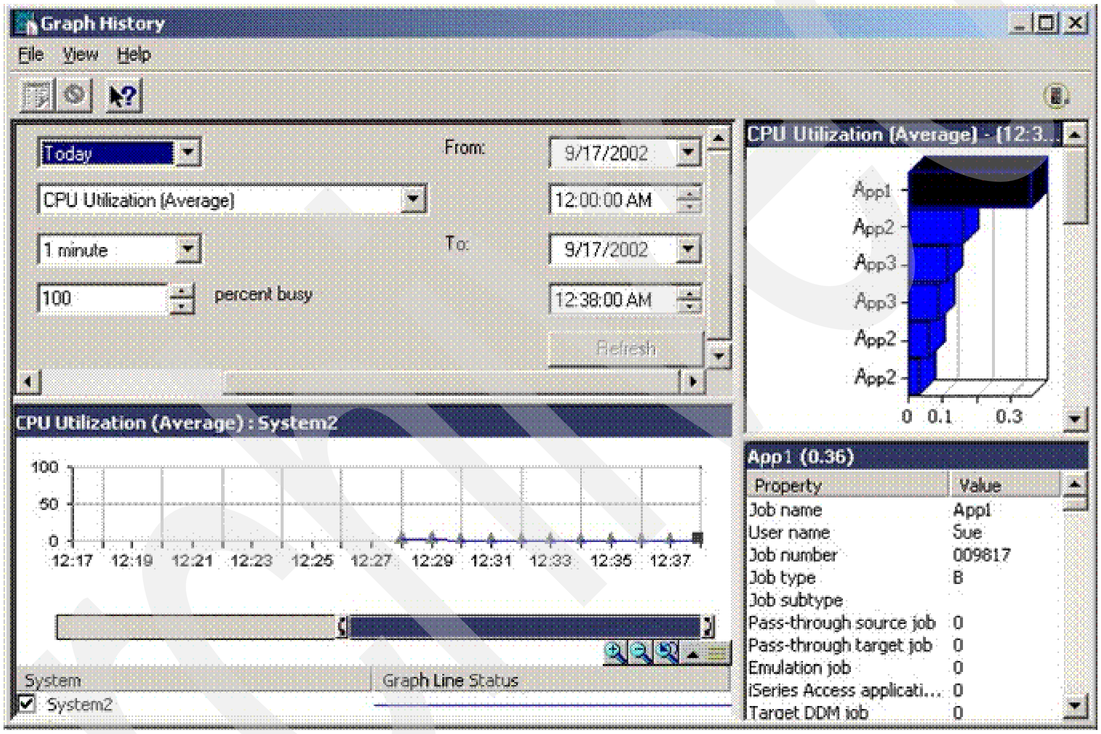
3.22.1 iSeries Navigator Graph History

The Graph History function is an extension of the System Monitors function. While system monitors display performance data only over the past hour, Graph History provides a graphical view of performance data collected over days, weeks, months, or years. It displays data in a format that is similar to the format provided by system monitors, but a system monitor does not have to be running to view the data. The source for Graph History data is a Management Collection object which is created and populated by Collection Services.

The amount of data that is available to be graphed is determined by the settings that are selected from the Collection Services properties, specifically the collection retention period. PM iSeries is required to be running on the system in order to view Graph History data that is older than seven days. You are not required to send PM iSeries data to IBM, but the PM iSeries collection facility needs to be running on the iSeries system.

PM iSeries reports and graphs provide additional detail that is not found in Graph History, so you might find it advantageous to send PM iSeries data to IBM to help you analyze performance and plan for future capacity needs.

The most common way to view Graph History data is to open the Configuration and Service task in the left pane of iSeries Navigator, right-click the Collection Services task, and select Graph History. Then you see a Graph History window like the example in Figure 3-26.



Today

CPU Utilization (Average)

1 minute

100 percent busy

From: 9/17/2002 12:00:00 AM

To: 9/17/2002 12:38:00 AM

☑ System2

App1 (0.36)

Property	Value
Job name	App1
User name	Sue
Job number	009817
Job type	B
Job subtype	
Pass-through source job	0
Pass-through target job	0
Emulation job	0
iSeries Access applicati...	0
Target DDM job	0

Figure 3-26 Graph History display

You can adjust the time frame, metrics, and frequency of the historical data that is displayed by adjusting the values in the upper left pane. You click the Refresh button in this pane to view the performance data which is displayed in the lower left pane. The detailed information in the right panes is displayed using the same techniques used for system monitors.

The amount of detailed data available in the right panes depends on the configuration options used by Collection Services. You right-click the Collection Services task in the left pane of iSeries Navigator and select Start Collection Services. From this window that opens, you can use the help text to learn how to configure Collection Services to provide the level of Graph and Summary data that you desire.

3.23 IBM Enterprise Workload Manager and ARM APIs

Modern Web-based applications typically run in a complex multi-tier environment, consisting of multiple edge-servers, routers, application servers, database servers, and so on. Execution of any single work request in this environment involves participation of multiple components, each of which impacts overall application performance. With so many components, it becomes increasingly difficult to manage application performance. If performance is not satisfactory, it becomes difficult to determine which component is responsible for performance degradation.

The system administrator might ask the following typical questions.

- Are work requests completing successfully? If not, where are they failing?
- Are successful work requests completing within the expected time? If not, where are the bottlenecks?
- How many work requests were completed over a period of time compared to prior periods? Is the workload growing?
- Are system-level resources optimally used? If not, can they be dynamically redirected elsewhere to achieve expected performance?

The purpose of IBM Enterprise Workload Manager (EWLM) is to address these kinds of problems in a world of distributed applications for On Demand Business. EWLM allows the application administrator to create a policy, which identifies different types of work that is performed in entire domain of servers in the enterprise. It also allows the administrator to assign different performance goals to each type of work. EWLM monitors performance of all these applications, compares real performance to performance goals set in a policy, and manages domain resources to ensure that performance goals are achieved.

In its monitoring role, EWLM has extensive reporting capabilities to provide real-time views of end-to-end performance in the enterprise network. In its first release, EWLM management capabilities include dynamic load-balancing of network applications. In future releases, EWLM will be able to dynamically balance logically partitioned (LPAR) resources in a partitioning environment. More management capabilities will be added as EWLM technology matures.

By its nature, EWLM is a multi-platform product. The EWLM management server is installed on one system in the enterprise domain. Every system or partition in the domain has the EWLM managed server installed. And every system or partition in the domain acts as an agent which supplies necessary information to the EWLM management server and performs resource management directives sent from the management server. EWLM has a version of the EWLM managed server for most major operating systems in the market today.

The key to EWLM functioning is the ability to identify work running on a system, to classify this work according to EWLM policy. It is also the ability to track the performance of work requests as they flow across the server and subsystem boundaries.

For this, EWLM relies heavily on Application Response Measurement (ARM) instrumentation of applications and middleware which glues applications together. ARM is an industry standard owned by The Open Group. The ARM standard includes a set of APIs, which are used by applications to make their identity and performance known to the underlying ARM implementation. EWLM uses version 4.0 of the ARM standard.

The basic concept in ARM is ARM transaction. ARM transaction may or may not have anything to do with other “transactions”, like database transactions. It is a responsibility of the application designer to decide which application work item to consider as a separate ARM transaction.

In a setup phase, an application calls ARM APIs to provide metadata about transactions performed by this application. This metadata is used by EWLM to identify work and classify it according to EWLM policy.

In a normal operating mode, an application calls ARM APIs to notify ARM implementation about the start and end of each application transaction. Applications use an ARM feature, called a *correlator*, to tie together pieces of multi-tier distributed transactions. A correlator is a unique token which must be passed by the application between different hops of a multi-tier transaction.

EWLM uses data provided by ARM implementation to reconstruct the application topology, the sequence of execution of complex distributed transactions, and to measure the real-time performance of ARM-instrumented applications.

IBM added ARM instrumentation to key middleware products that it ships, most notably the IBM HTTP Server (powered by Apache), WebSphere Application Server, and DB2® database server. The process of adding ARM instrumentation to other IBM and non-IBM applications and middleware products will continue into the future.

EWLM is a component of the IBM Virtualization Engine™ product. You can learn more about EWLM and the usage of ARM APIs in EWLM section of the IBM @server Software Information Center.

http://publib.boulder.ibm.com/infocenter/eserver/v1r1/en_US/info/ewlminfo/kickoff.htm

You can also find more information about ARM APIs on the Open Group Web site at:

http://www.opengroup.org/arm/

3.24 IBM Director

IBM Director Resource Monitors are a set of system monitors that you can run from the IBM Director Console. They monitor resource usage from systems that are running Director Agent. This is a convenient way to monitor system metrics, but it takes advance work to use it. IBM Director Console must be installed on your PC, and the Director Agent must be installed on every machine (Linux, AIX, iSeries) from which you want to see data.

3.24.1 How to use IBM Director

Launch the Director Console on your PC and specify a system on which IBM Director Server is running. Then you see a screen divided into three columns: Groups, Group Contents, and Tasks like the example in Figure 3-27.

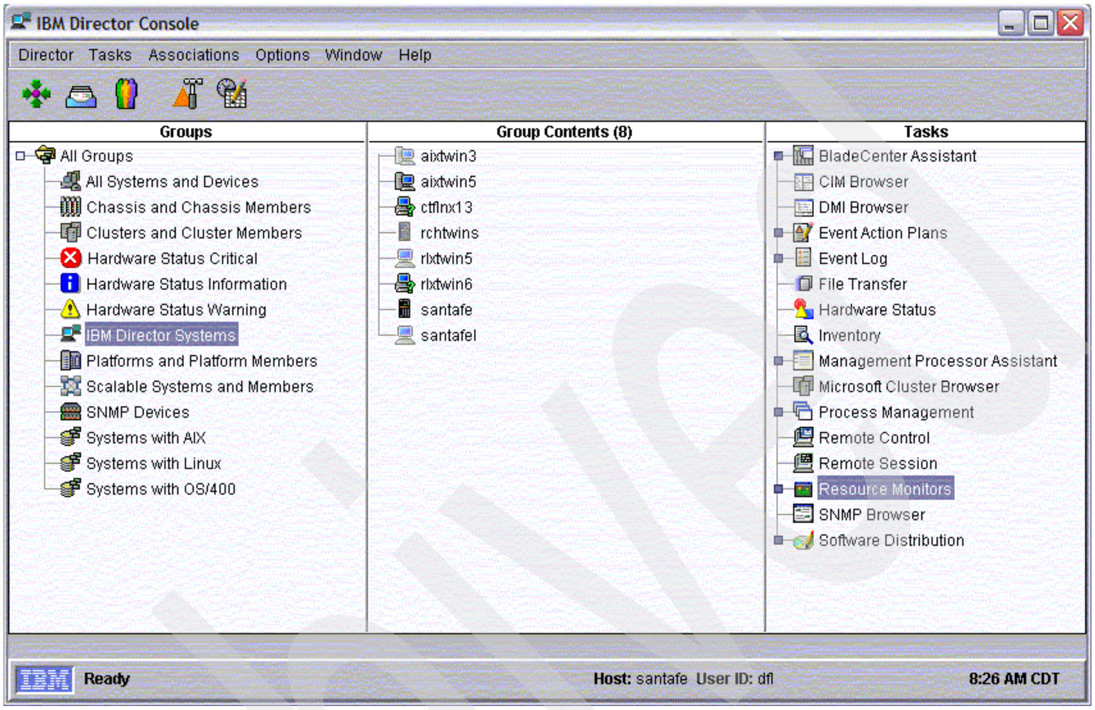
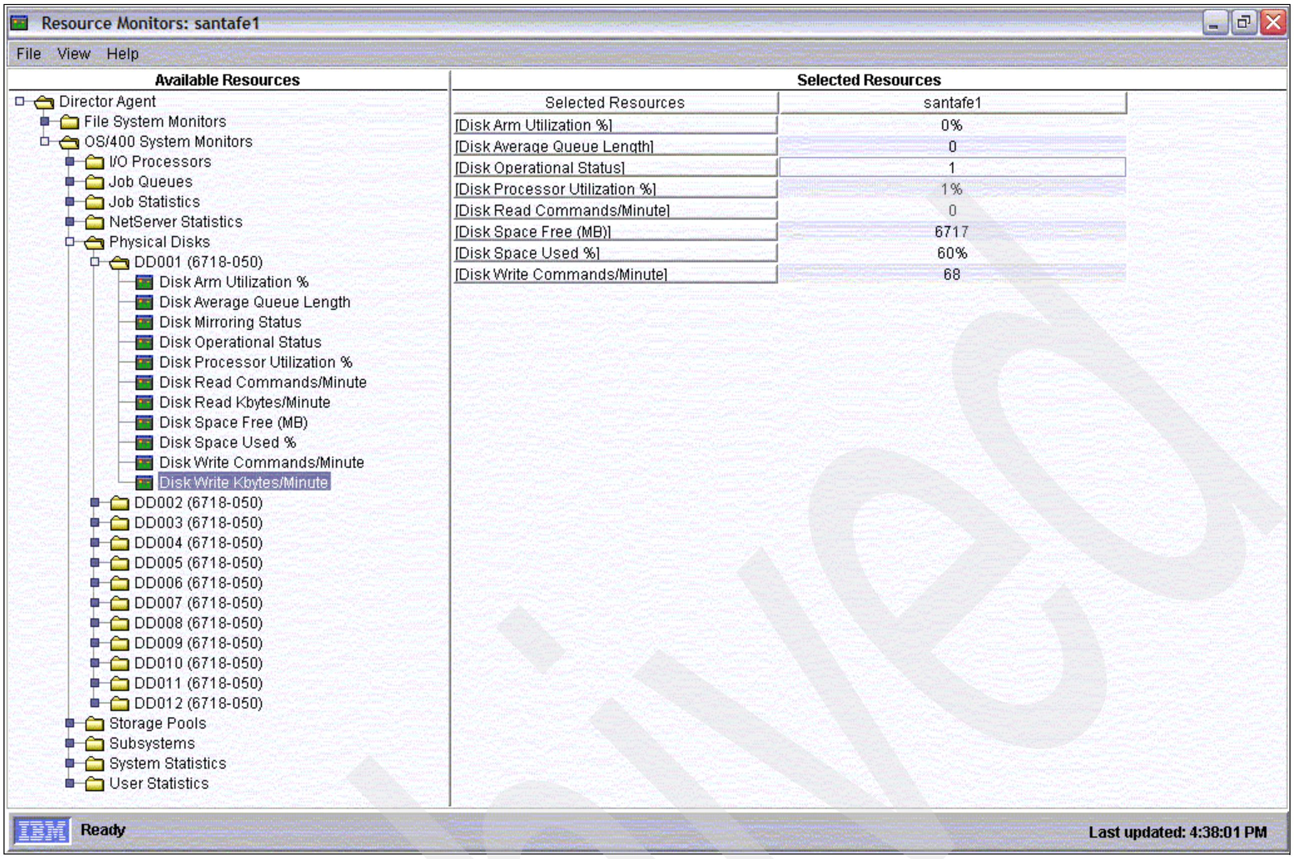


Figure 3-27 IBM Director Console

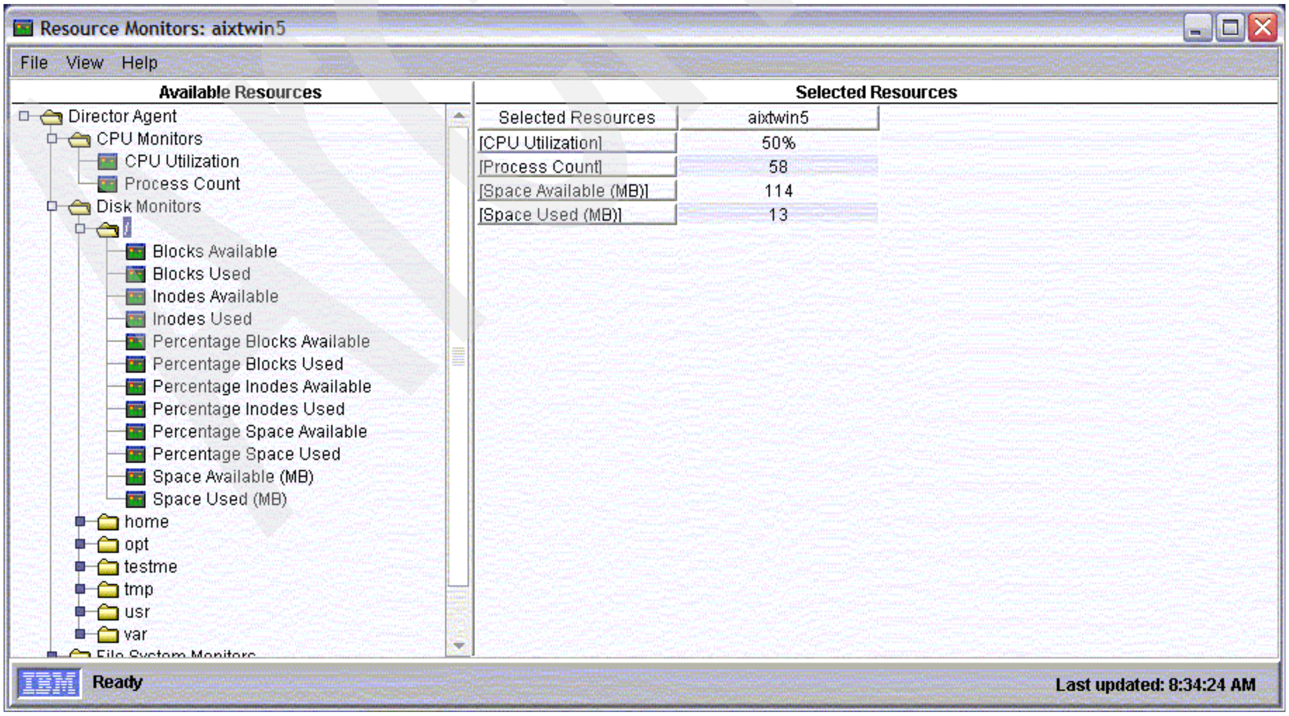
Drag the Resource Monitor task on top of a system in the Group Contents column to run Resource Monitors on that system. Then you see a list of hardware resources that you can expand. Continue expanding the items until you reach the hardware as shown in Figure 3-28.



Selected Resources	santafe1
[Disk Arm Utilization %]	0%
[Disk Average Queue Length]	0
[Disk Operational Status]	1
[Disk Processor Utilization %]	1%
[Disk Read Commands/Minute]	0
[Disk Space Free (MB)]	6717
[Disk Space Used %]	60%
[Disk Write Commands/Minute]	68

Figure 3-28 Sample of Resource Monitor output from i5/OS

Figure 3-29 shows an example of output from a system with an AIX operating system.



Selected Resources	aixtwin5
[CPU Utilization]	50%
[Process Count]	58
[Space Available (MB)]	114
[Space Used (MB)]	13

Figure 3-29 Sample of Resource Monitor output from AIX

Find more information about IBM Director, including a link to download it, at the following Web site.

http://www-03.ibm.com/servers/eserver/xseries/systems_management/director_4.html

3.25 Performance Trace Data

Performance Trace Data is a special type of trace data that is customized for several reports in the Performance Tools LPP. It contains multiprogramming level (MPL) trace data, transaction trace data, and resource management trace data (seizes and locks). This trace is best suited for analyzing jobs that run 5250-type interactive transactions. It does not have the customization options of the PEX *TRACE support. Unlike the interval data produced by Collection Services, which provides information at the job level, trace data lets you analyze individual transactions.

Use the STRPFRTRC command in i5/OS to start collection of Performance Trace data. This command is actually a simplified interface to the Trace Internal (TRCINT) command for collecting MPL Trace data. The trace table size that is used is set automatically, based on the size of your machine. If you want to ignore seize and lock data, specify OMTTRCPNT(*RSCMGMT). This can be useful on a system with lots of short seizes and locks which you do not want to analyze. Omitting them from the collection allows the trace table to hold more non-resource management trace data.

Use the ENDPFRTRC command in i5/OS to dump the data into the QAPMDMPT database file. You specify the library name and the member name.

To analyze the collected data, use the PRTTNSRPT and PRTLCKRPT commands in the Performance Tools Licensed Program Product.

You can use the Transaction report (via the PRTTNSRPT command) as a starting point to analyze information about 5250-type interactive transactions, such as poor transaction response time, CPU time used, number of I/O requests, and seize/lock wait times. You can use the Interactive Program Statistics section in the Job Summary report (PRTTNSRPT RPTTYPE(*SUMMARY)) to identify the top 10 programs with the largest average CPU time per transaction, disk I/O per transaction, response time per transaction, and database/nondatabase reads/writes per transaction.

You can use the Lock report (via the PRTLCKRPT command) to see the lock and seize conflicts that were captured in the resource management trace data. You can use this information to determine if jobs are delayed during processing because of unsatisfied lock requests or internal machine seizes (internal waits).

During the analysis of jobs with long delays, keep in mind that those jobs can be the effect and not the cause of a performance problem caused by object contention.

3.26 Performance Explorer

PEX consists of a collection mechanism (ADDPEXDFN, STRPEX, and ENDPEX commands) built into the operating system and a print command (PRTPEXRPT), which is part of the Performance Tools Licensed Program Product. Four types of PEX data can be collected:

- *TRACE
- *STATS
- *PGM/*PROFILE
- *JOB/*PROFILE (also known as TPROF)

You can use PEX to collect higher-level information (*STATS and *JOB/PROFILE) as a first step in getting details about your system's CPU usage. And you can use it to collect detailed, trace-level data (*TRACE), which can be used to determine causes of performance problems that cannot be identified by other tools. You can also use *PGM/PROFILE to analyze individual programs to find high CPU usage areas within the program either during code development or on a production system.

3.26.1 How PEX works

STRPEX communicates with the Performance Data Collector (PDC) running in SLIC, which performs the actual data collection. ENDPEX tells PDC to end the collection, and then retrieves the PDC data and writes it into a set of database files which can be analyzed later by PRTPEXRPT or Performance Trace Data Visualizer (PTDV). See 3.28, "Performance Trace Data Visualizer" on page 63.

3.26.2 How to use PEX

Specify what you want to do by creating a PEX definition using the ADDPEXDFN command. This is where you specify the type of the collection, specify jobs you want to collect data for, or the trace events you want.

To start the collection, run the STRPEX command and provide the PEX definition that you previously created. To end the collection, run the ENDPEX command. To print the data, use the PRTPEXRPT command. In most cases, you can also use the PTDV tool to analyze the data. See 3.28, "Performance Trace Data Visualizer" on page 63, for details.

Table 3-2 explains the different types of PEX collections.

Table 3-2 Different types of PEX collections

Collection type	Disk overhead	CPU overhead	Used for
*STATS	Low	Medium/ High	Call/return data for OPM programs and "hooked" ILE programs. Create definition with ADDPEXDFN TYPE(*STATS)... Analyze with PRTPEXRPT TYPE(*STATS)... or PTDV
*JOB/ PROFILE (TPROF)	Medium	Very Low	Histogram of CPU usage across entire system; trace collection printed as profile Create definition with ADDPEXDFN TYPE(*PROFILE) PRFTYPE(*JOB)... Analyze with PRTPEXRPT TYPE(*TRACE).... or PTDV
*PGM/ PROFILE	Low	Medium	Histogram of CPU usage for an individual program; use after STATS or TPROF Create definition with ADDPEXDFN TYPE(*PROFILE) PRFTYPE(*PGM)... Analyze with PRTPEXRPT TYPE(*PROFILE)...
*TRACE	High	Low	Trace selected events; don't use this as the first step. Create with definition ADDPEXDFN TYPE(*TRACE)... Analyze with PRTPEXRPT TYPE(*TRACE)... or PTDV

3.27 iDoctor for iSeries PEX Analyzer

The iDoctor for iSeries PEX Analyzer tool facilitates the analysis of PEX data. PEX Analyzer is a software tool that is specifically tuned for identifying issues affecting system and application performance. It provides detailed analysis that goes beyond the detail provided by PM/400 and Performance Tools. It supplies a drill-down capability that provides a low-level summary of disk operations, CPU utilization, file opens, MI programs, wait states, disk space consumption and much more.

PEX Analyzer allows a user to create, delete, copy, save, use File Transfer Protocol (FTP), and analyze PEX collections as well as create, change or delete PEX definitions. The tool also provides graphical visualization of PEX collection data. This helps the user to quickly identify performance problems by condensing and graphing PEX trace, statistical, and profile data. In the past, the typical method for creating reports over PEX data was to use the Print PEX Report (PRTPEXRPT) command. PEX Analyzer is a value-add to this command by providing more flexible interfaces for viewing a potentially large amount of complex data.

PEX Analyzer works with PEX trace data previously collected on the current or other iSeries system. PEX Analyzer can only analyze data that has been collected. For example, if the PEX trace collection did not include database file activity, PEX Analyzer cannot produce reports based on database file activity. It is important to understand that it is at the time of PEX data collection that the decision is made concerning which types of data are collected. This decision controls the type of data which will be available for analysis.

For V5R3 documentation of iDoctor for iSeries, go to the following Web site.

https://www-912.ibm.com/i_dir/idoctor.nsf/F204DE4F34767E0686256F4000757A90/$FILE/iDoctorV5R3Help.pdf

3.28 Performance Trace Data Visualizer

The PTDV for iSeries tool is used to process, analyze, and view Performance Explorer Collection data residing in PEX database files with an emphasis on the PEX events used in Java programs. This tool was designed for analysis of Java applications on the iSeries, differentiating it from PEX Analyzer. Therefore it is limited to collections which are meaningful in a Java environment. These collections include PEX *TRACE collections containing program, Java, or OS events for WebSphere.

PTDV provides both table and tree views of PEX trace data. It summarizes this data on various levels, providing many views and various levels of detail depending on your path through the data. Tables consist of columns which are sortable and rearrangeable, and trees provide a drill-down capability to show the parent/child relationship for data where appropriate. This feature is valuable when viewing the tree structure of a call trace. Each pane of data can be exported to another format for viewing by other tools, such as spreadsheets. As the user determines performance problem areas, more detail can be displayed.

The following views are provided by PTDV.

- Job and thread summary information is provided and includes cycle counts, elapsed times, and instruction counts for each job and thread. The collection data for an individual job or thread can also be viewed.
- When the appropriate Java events are present in the PEX collection, object summary information is provided. This includes object creation and lock details that show which types of objects have excessive creates or long wait or hold times.

- When the appropriate program events are included in the PEX collection, procedure/method summary information is provided. This includes the times called, inline and cumulative time, and cycles. This information can help to determine which procedures or methods are using the most time. If there are Java events in addition to program events in the collection, object information is summarized on a method basis.
- When program events are found in the collection, a call trace is built in tree format, including elapsed time and cycle counts for each individual call. If Java events are included in the collection, the Java event information is summarized on a per method call basis.

In addition to support for PEX *TRACE collections, PTDV processes, analyzes, and displays information based on the data in a Trace/Profile (TPROF) collection. This option can provide details from the PEX data that are not seen with PRTPEXPRT. You can select significant subsets of data and drill down to find more details about the important information in the collection.

Find the V5R3 documentation of iDoctor for iSeries on the Web at:

https://www-912.ibm.com/i_dir/idoctor.nsf/F204DE4F34767E0686256F4000757A90/$FILE/iDoctorV5R3Help.pdf

3.29 Database Monitor for iSeries

The Database Monitor, a part of i5/OS, collects database performance statistics for a specified job or for all jobs on the system. You can use the output to analyze database performance problems after SQL requests have been run. To start the monitor, run the STRDBMON command and specify the job to collect (or *ALL) and the file which will receive the output records. Use the ENDDBMON command to end the collection of data.

The data collected via the GUI is the same as the data collected via a 5250 session. The data collected via the GUI is called *SQL performance monitor data*.

Refer to Chapter 4 about using performance tools for determining performance problems in the IBM Redbook *Maximum Performance with WebSphere Application Server V5.1 on iSeries*, SG24-6383.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 65. Note that some of the documents referenced here may be available in softcopy only.

- *Maximum Performance with WebSphere Application Server V5.1 on iSeries*, SG24-6383
- *IBM iDoctor for iSeries Job Watcher: Advanced Performance Tool,* SG24-6474

Other publications

These publications are also relevant as further information sources:

- *IBM @server iSeries Performance Capabilities Reference*, SC41-0607

  http://www.ibm.com/eserver/iseries/perfmgmt/resource.html
- *Performance Tools for iSeries*, SC41-5340-01

Online resources

These Web sites and URLs are also relevant as further information sources:

- iSeries Information Center performance-based articles and PDFs under **Systems Management** → **Performance**

  http://www.ibm.com/eserver/iseries/infocenter
- iSeries Performance Management articles and PDFs

  http://www.ibm.com/eserver/iseries/perfmgmt

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM eserver iSeries Performance Management Tools

Redpaper

Learn about the various iSeries performance management tools

Find answers to common performance questions matched to the right tools

Explore details about why to use each tool

Learn about the complete array of IBM eserver iSeries performance management tools! This IBM Redpaper is designed to help you understand the different iSeries performance management tools at the IBM i5/OS V5R3M0 level, that are available to you and when to use them. The paper is designed for beginners who want to learn about the tools that are available. It also provides a helpful reference to the more experienced user who needs one resource for all the performance management tools available to them.

The paper begins with a comprehensive list of the tools. Then the paper presents common questions that clients ask when managing performance. Each question is matched with the recommended tool or tools to use to help you determine the answer. Finally the paper provides details about why you should use a particular tool and links to more information about obtaining and using each one.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks