CICS Performance Analyzer

Gain insight to the IBM WebSphere MQ reporting capabilities

Explore enhanced CICS Monitoring Facility (CMF) resource class reporting

Learn how to work with historical performance data

Chris Rayns
Shirley Collins
Eugene Deborin
Peter Klein
Greg Todaro
Mark Todd
Norbert Verbestel

ibm.com/redbooks
Note: Before using this information and the product it supports, read the information in “Notices” on page xi.
# Contents

Notices ................................................................. xi  
Trademarks .............................................................. xii  

Preface ................................................................... xiii  
The team who wrote this book ......................................... xiii  
Become a published author ............................................. xiv  
Comments welcome .................................................... xv  

Part 1. CICS Performance Analyzer overview ......................... 1  

Chapter 1. CICS performance management ............................ 3  
1.1 How to approach CICS performance monitoring and tuning .... 4  
1.2 CICS tools ............................................................... 4  
1.2.1 CICS Performance Analyzer .................................... 5  
1.2.2 CICS Performance Monitor ...................................... 6  
1.2.3 Benefits of CICS Performance Analyzer and CICS Performance Monitor ........................................ 7  
1.2.4 CICS Online Transmission Time Optimizer ................. 8  
1.2.5 CICS Interdependency Analyzer ............................... 8  
1.2.6 CICS Business Event Publisher for MQSeries ............ 8  
1.2.7 IBM Session Manager for z/OS ................................ 9  
1.2.8 CICS VSAM Recovery ........................................... 9  
1.3 Data used by CICS Performance Analyzer ...................... 10  
1.3.1 CICS Monitoring Facility data (SMF 110 records) ......... 10  
1.3.2 DB2 accounting data (SMF 101 records) .................... 14  
1.3.3 WebSphere MQ accounting data (SMF 116 records) .... 15  
1.3.4 MVS System Logger data (SMF 88 records) ............... 16  
1.4 Other relevant CICS data and utilities ........................... 17  
1.4.1 CICS statistics ...................................................... 18  
1.4.2 The sample statistics program: DFHHOSTAT ............... 19  
1.4.3 Statistics utility program: DFHSTUP ......................... 19  
1.4.4 Monitoring dictionary utility program: DFHMNDUP ....... 20  
1.4.5 Sample monitoring data print program: DFH$MOLS ........ 20  
1.4.6 Journal utility program: DFHJUP ............................ 21  
1.5 Other relevant information sources .............................. 21  
1.5.1 System Management Facility .................................. 21  
1.5.2 Resource Management Facility ................................. 22  
1.5.3 Generalized Trace Facility ...................................... 22  
1.5.4 Tivoli Decision Support for OS/390 ......................... 22  

Chapter 2. A quick start to CICS Performance Analyzer .............. 23  
2.1 What CICS PA is ...................................................... 24  
2.2 System requirements ............................................... 25  
2.3 Recommended ISPF setup .......................................... 26  
2.4 Preparing the SMF data for CICS PA ............................ 28  
2.5 Starting CICS PA ...................................................... 29  
2.5.1 CICS PA Primary Option Menu ............................... 29  
2.6 System definitions .................................................. 29  
2.6.1 System Definitions Menu ...................................... 30  
2.6.2 Take-Up from SMF .............................................. 30
Chapter 6. VSAM application performance analysis and Transaction Resource Monitoring support

6.1 CICS VSAM interface .................................................. 140
6.2 CICS VSAM RLS scenario description
   6.2.1 RLS workload generation .................................. 141
   6.2.2 Performance objectives .................................. 142
6.3 Running the VSAM RLS scenario .................................. 143
   6.3.1 Updating system definitions .............................. 143
   6.3.2 Creating a Summary Report Form ...................... 145
   6.3.3 Creating an Object List ................................. 146
   6.3.4 Creating a Report Set ................................ 147
6.4 CICS VSAM LSR scenario description ............................ 157
6.5 Running the VSAM LSR scenario .................................. 157
   6.5.1 LSR workload generation ................................ 157
   6.5.2 LSR application performance objectives ............. 158
   6.5.3 Analyzing the current average response time ....... 158
   6.5.4 Tuning changes to LSR ................................. 161
   6.5.5 Analyzing results ...................................... 162
6.6 Application performance RLS versus LSR ......................... 163
6.7 Transaction Resource Monitoring .................................. 163
6.8 Conclusion .......................................................... 169

Chapter 7. Tuning the CICS DB2 attachment facility

7.1 CICS and DB2 .......................................................... 172
   7.1.1 Overview of the CICS DB2 Attachment Facility .... 172
   7.1.2 Functions .................................................... 172
   7.1.3 CICS and DB2 connectivity ............................. 173
7.2 CICS DB2 accounting and monitoring .............................. 177
7.3 CICS PA reporting CICS DB2 Attachment Facility ............... 179
7.4 CICS PA example reports ........................................ 179
7.5 CICS TS V1.3 and DB2 V7 scenario ............................... 187
7.6 CICS TS V2.2 and DB2 V7 scenario ............................... 194
7.7 Extracting CICS DB2 records .................................... 200
7.8 Conclusion .......................................................... 203

Chapter 8. WebSphere MQ

8.1 Overview ............................................................. 206
8.2 Environment .......................................................... 206
8.3 CICS changes .......................................................... 207
8.4 MQ accounting trace ............................................... 207
8.5 MQ reports ............................................................ 209
   8.5.1 MQ class 1 reports ....................................... 210
   8.5.2 MQ Class 3 reports ....................................... 213
   8.5.3 Performance List report ................................ 216
   8.5.4 Transaction Resource Usage Temporary Storage report ............................................... 218
8.6 Conclusion .......................................................... 221

Chapter 9. CICS and MVS System Logger

9.1 CICS TS and the MVS System Logger ............................. 224
9.2 CICS PA reporting on the System Logger .......................... 226
9.3 Scenario description ............................................... 226
9.4 Scenario run .......................................................... 227
9.5 Conclusion .......................................................... 239
**Chapter 10. Scenarios with CICS Transaction Gateway**

10.1 What is CICS Transaction Gateway .......................... 242
   10.1.1 Gateway components and downstream protocols ........ 242
   10.1.2 Application programming interfaces and upstream protocols .... 243
10.2 Scenario description ........................................ 244
   10.2.1 Front-end application .................................... 245
   10.2.2 Back-end application ..................................... 245
   10.2.3 CTG for z/OS EXCI scenario .............................. 246
   10.2.4 Distributed CTG using ECI over TCP/IP ................. 246
10.3 Scenario run .................................................. 246
   10.3.1 Running the workload using the CTG for z/OS ............ 247
   10.3.2 Running the workload using the distributed CTG ......... 252
10.4 Conclusion ....................................................... 256

**Chapter 11. CICS Web Support and 3270 Bridge**

11.1 CICS Web Support overview ................................... 258
11.2 Scenario description .......................................... 258
11.3 Scenario run .................................................... 260
11.4 Tuning your CICS Web Support environment .................... 263
   11.4.1 Storage consumption by CWXN ............................ 263
11.5 Conclusion ....................................................... 266

**Chapter 12. Java applications in CICS**

12.1 CICS and Java .................................................. 268
   12.1.1 Java language programs in CICS ......................... 268
   12.1.2 Stateless CORBA objects .................................. 268
   12.1.3 Enterprise JavaBeans ...................................... 268
12.2 Scenario description .......................................... 269
12.3 Measuring JVM performance .................................... 271
   12.3.1 Xresettable=NO ............................................ 271
   12.3.2 Xresettable=YES .......................................... 279
   12.3.3 Using the sharable application classpath ............... 281
12.4 Conclusion ....................................................... 283

**Chapter 13. Enterprise JavaBeans in CICS**

13.1 CICS logical EJB server configuration ......................... 286
13.2 Single thread enterprise bean scenario ....................... 287
   13.2.1 CICS TS V2.2 JDBC/SQLJ support ....................... 287
   13.2.2 Initiating transaction workload ......................... 287
13.3 Multithread enterprise bean scenario ........................ 300
   13.3.1 Workload generation ..................................... 300
   13.3.2 Objectives ............................................... 302
   13.3.3 Running the SQLJ workload ............................. 302
   13.3.4 Running the JDBC workload ............................. 305
   13.3.5 Running the JCICS (VSAM) workload .................... 307
13.4 Conclusion ....................................................... 310

**Chapter 14. Application Naming support**

14.1 Implementing Application Naming support .................... 312
14.2 Application Naming sample ................................... 312
   14.2.1 Sample program .......................................... 312
   14.2.2 Monitoring control table ............................... 314
   14.2.3 Program run results using the Performance List report ... 315
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.6 Producing the Cross-System Work report</td>
<td>403</td>
</tr>
<tr>
<td>18.7 Producing the File Usage detailed list report</td>
<td>406</td>
</tr>
<tr>
<td>18.8 The second CSMI mirror task</td>
<td>409</td>
</tr>
<tr>
<td>18.9 Options to improve transaction BR1 performance</td>
<td>411</td>
</tr>
<tr>
<td>18.9.1 Define the VENDORX file as a CICS Maintained Data Table</td>
<td>411</td>
</tr>
<tr>
<td>18.9.2 Define the VENDORX file locally</td>
<td>412</td>
</tr>
<tr>
<td>18.10 Future problem determination efforts</td>
<td>412</td>
</tr>
<tr>
<td>18.11 Conclusion</td>
<td>413</td>
</tr>
<tr>
<td>Chapter 19. Historical Database</td>
<td>415</td>
</tr>
<tr>
<td>19.1 Introduction to the HDB</td>
<td>416</td>
</tr>
<tr>
<td>19.2 System setup and scenario overview</td>
<td>416</td>
</tr>
<tr>
<td>19.3 CICS changes</td>
<td>417</td>
</tr>
<tr>
<td>19.4 Using the HDB</td>
<td>418</td>
</tr>
<tr>
<td>19.4.1 Summary HDB</td>
<td>418</td>
</tr>
<tr>
<td>19.4.2 List HDB</td>
<td>437</td>
</tr>
<tr>
<td>19.4.3 HDB maintenance</td>
<td>437</td>
</tr>
<tr>
<td>19.4.4 HDB housekeeping</td>
<td>439</td>
</tr>
<tr>
<td>19.4.5 Export HDB data sets to DB2</td>
<td>441</td>
</tr>
<tr>
<td>19.5 Conclusion</td>
<td>451</td>
</tr>
<tr>
<td>Abbreviations and acronyms</td>
<td>453</td>
</tr>
<tr>
<td>Related publications</td>
<td>455</td>
</tr>
<tr>
<td>IBM Redbooks</td>
<td>455</td>
</tr>
<tr>
<td>Other publications</td>
<td>455</td>
</tr>
<tr>
<td>Online resources</td>
<td>456</td>
</tr>
<tr>
<td>How to get IBM Redbooks</td>
<td>456</td>
</tr>
<tr>
<td>Help from IBM</td>
<td>456</td>
</tr>
<tr>
<td>Index</td>
<td>457</td>
</tr>
</tbody>
</table>
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 illustrate 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.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

1-2-3®  Language Environment®  Redbooks (logo) ®
AIX®  Lotus®  Resource Measurement Facility™
AnyNet®  MQSeries®  RMF™
CICSPlex®  MVS™  S/390®
CICS®  OS/390®  Tivoli®
DB2®  Parallel Sysplex®  VTAM®
IBM®  RACF®  WebSphere®
iSeries®  Redbooks®  z/OS®

The following terms are trademarks of other companies:

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

Microsoft, Windows NT, Windows, 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, or service names may be trademarks or service marks of others.
Preface

This IBM® Redbooks® publication targets CICS® Transaction Server V1.3 and V2.2 customers who plan to implement IBM CICS Performance Analyzer. With this tool, you can produce a wide range of reports and extracts to help you tune and manage CICS systems.

The first part of this book begins with an overview of CICS-provided tools and utilities that help you gather and analyze performance data. Then it introduces the CICS Performance Analyzer product and its various report generating options. It also shows you how to import the extracted performance data into spreadsheets for further analysis.

The second part of this book takes you through a series of scenarios that cover major CICS components and interfaces. These include CICS-VSAM interface, CICS-DB2 Attachment Facility, CICS use of the MVS™ System Logger, Java™ applications in CICS, and others. For each scenario, you see how you can extract the relevant performance data using CICS Performance Analyzer. You can then use this data to improve the overall system performance or to compare different execution options at run time.

This Redbooks publication explores the new functionality of CICS PA Release 1.3, including IBM WebSphere® MQ and how CICS PA now handles System Management Facility (SMF) 116 records. It looks at the new CICS Monitoring Facility (CMF) reports such as the Wait Analysis and Temporary Storage Usage reports. It also explains the Historical Database (HDB) facility for maintaining a history of CMF performance data for longer term reporting or exporting to DB2®.

Note: This book is based on the Redbooks publication IBM Tools: CICS Performance Analyzer V1.2, SG24-6882.

The team who wrote this book

This Redbooks publication was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

Chris Rayns is an IT Specialist and project leader at the ITSO, Poughkeepsie Center in New York. Specializing in security, Chris writes extensively on all areas of S/390® security. Before joining the ITSO, he worked in IBM Global Services in the United Kingdom (UK) as a CICS IT Specialist.

Shirley Collins is a Technical Writer with Fundi Software in Perth, Western Australia. She is a member of the development team for CICS Performance Analyzer, IMS™ Performance Analyzer, and IMS Problem Investigator. Prior to this position, she was with IBM Australia after starting out as a programmer/analyst on financial systems. She holds degrees in mathematics and education.

Eugene Deborin is a Consulting Information Technology Specialist at the ITSO, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide about all areas of transaction and message processing. Eugene has been working for IBM Israel since 1984, first as a systems engineer and, more recently, as a software technical sales specialist. He joined the ITSO San Jose Center in 1997 as a CICS Transaction Server Specialist and is now a CICS and WebSphere MQ project leader at the ITSO, Raleigh Center in North Carolina.
Peter Klein is a CICS Team Leader at the IBM Germany Customer Support Center. He has 18 years of experience working as a technical support specialist with IBM software products. His expertise includes WebSphere MQ, CICSPlex® System Manager, and distributed transaction systems. Peter has contributed to several other Redbooks publications and ITSO projects sponsored by IBM Learning Services.

Greg Todaro is an IBM Software Engineer with the CICS/390 Level 2 support group in Raleigh, North Carolina. Greg began working with CICS in 1985 as a programmer/analyst. Then, he moved quickly into MVS and CICS systems programming for the insurance, banking, and manufacturing sectors. He has been with IBM since 1998.

Mark Todd is a Software Engineer from IBM Hursley, England. He has been on the CICS/390 Change Team, and with IBM, for five years. He also has 13 years of experience in working as a systems programmer, for IBM software products, in the manufacturing and financial sectors.

Norbert Verbestel is an Information Technology Specialist at the IBM Belgium Support Center. His areas of responsibility include second-level support, customer service, and education. He has been with IBM for 28 years, spending 21 of those years working with the CICS product. His other areas of expertise are DB2 and CICSPlex System Manager. Norbert has contributed to several other Redbooks publications and ITSO projects sponsored by IBM Learning Services.

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

Jim Grauel
IBM US, CICS Level 2 Support

Paola Bari
Richard Conway
Robert Haimowitz
IBM ITSO, Poughkeepsie Center

Bart Steegmans
IBM ITSO, San Jose Center

Chris Baker
Phil Wakelin
IBM UK, Hursley Laboratories

John Hancy
Fundi Software, Australia

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 Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online Contact us review Redbooks 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. HYJ  Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400
Part 1 offers a theoretical look at CICS Performance Analyzer. It introduces the CICS Performance Analyzer product, a component of the IBM CICS Tools family. It explains how CICS reports performance information. Then it takes you through the main CICS Performance Analyzer menus and options. This part presents an overview of performance reports and extracts that CICS Performance Analyzer can generate. It also shows how you can process extracts using a variety of tools.

Part 2 presents a practical look at CICS Performance Analyzer. It takes you through actual scenarios. It shows you how to run CICS Performance Analyzer reports and extracts to analyze system performance. Part 2 also discusses the CICS Performance Analyzer Historical Database (HDB).
This chapter discusses the basics of CICS performance monitoring and tuning methodology. It introduces two complementary CICS performance tools to help you analyze and improve the performance of your CICS systems:

- CICS Performance Analyzer (CICS PA)
- CICS Performance Monitor (CICS PM)

It also mentions the following CICS tools:

- CICS Online Transmission Time Optimizer
- CICS Interdependency Analyzer
- CICS Business Event Publisher for MQSeries®
- IBM Session Manager for z/OS®
- CICS VSAM Recovery

This chapter describes the tools that CICS itself provides to help you gather data that serves as input to CICS Performance Analyzer. Plus it describes CICS statistics processing and monitoring utilities that you may find useful to use in conjunction with CICS Performance Analyzer.
1.1 How to approach CICS performance monitoring and tuning

CICS performance management is the process of continuously monitoring, analyzing, and improving the behavior of your system so that you meet the service levels that you are committed to. Good performance is the achievement of the agreed service levels. It means that system throughput, system availability, and response times meet user's expectations using resources within the budget.

There are several basic steps in tuning a system, some of which may be iterative:

1. Set up performance objectives.
   Performance objectives often consist of a list of transactions and expected response times for each. Ideally, through them, good performance can be easily recognized. Therefore, they must be:
   - Practically measurable
   - Based on a realistic workload
   - Within the budget
   
   The performance objectives must be agreed upon and regularly reviewed with the users.

2. Decide on measurement criteria.
   Performance objectives may be defined in such terms as:
   - Desired or acceptable response times, such within which 80% of all responses occur
   - Average or peak number of transactions through the system
   - System availability, including mean time to failure and downtime after failure

3. Gather the performance data of your production system.
   CICS provides a variety of tools that help you gather performance data for online monitoring or statistical analysis.

4. Analyze this performance data.
   Use the online performance monitoring and the offline performance reporting tools and apply the methodology described in Part 3 of CICS Transaction Server for z/OS CICS Performance Guide, SC34-6009.

5. Adjust the system as necessary.

6. Continue to monitor the performance of the system and anticipate future constraints.

1.2 CICS tools

In recent years, in response to customer requirements, IBM has developed an extensive portfolio of tools for use by customers running CICS Transaction Server on OS/390® or z/OS. Currently, the CICS tools portfolio includes:

- CICS Performance Analyzer
- CICS Performance Monitor
- CICS Online Transmission Time Optimizer
- CICS Interdependency Analyzer
- CICS Business Event Publisher for MQSeries
- IBM Session Manager for z/OS
- CICS VSAM Recovery
These tools complement the comprehensive range of IBM @server zSeries® tools:

- Data Management Tools (for example, IBM DB2 Performance Monitor for OS/390, IBM DB2 SQL Performance Analyzer for OS/390)
- Application Development Tools (for example, IBM Fault Analyzer for z/OS and OS/390, IBM File Manager for z/OS and OS/390, Debug Tool for z/OS and OS/390)
- System Management and other tools (for example, CICS VSAM Recovery)

Together with these other tools from IBM, CICS tools provide customers with an opportunity to significantly reduce the total cost of ownership of their z/OS and OS/390 systems.

### 1.2.1 CICS Performance Analyzer

CICS PA is a reporting tool that provides information about the performance of your CICS systems and applications. It helps you tune, manage, and plan your CICS systems in an efficient way.

CICS PA provides a Historical Database (HDB) facility to help you manage the performance data for your CICS transactions.

CICS PA provides reports and extracts using the data that is normally collected by your system in system management facility (SMF) data sets:

- CICS Monitoring Facility (CMF) performance, exception, and transaction resource class records (type 110)
- DB2 accounting records (type 101)
- WebSphere MQ accounting records (type 116)
- System Logger records (type 88)

CICS PA is designed to complement the CICS-supplied utilities and sample programs, such as DFH$MOLS, DFHSTUP, and DFH0STAT.

CICS PA can help:

- System programmers to track overall CICS performance and evaluate the results of their system tuning efforts
- Application programmers to analyze the performance of their applications and the resources they use
- Database administrators to analyze the usage and performance of database systems such as IMS and DB2
- MQ administrators to analyze the usage and performance of their WebSphere MQ messaging systems
- Managers to ensure that their service-level agreement objectives are met and measure trends to help plan future requirements

CICS PA reports all aspects of CICS system activity and resource usage, including:

- Transaction response time and resource usage
- CICS system resource usage
- Cross-system performance, including multiregion operation (MRO) and advanced program-to-program communication (APPC)
- Business Transaction Services (BTS)
- CICS Web Support (CWS)
- External subsystems, including DB2, IMS, and WebSphere MQ
- System Logger performance
- Exception events that cause performance degradation

CICS PA provides both an ISPF screen and a command interface. You can use either to request your reports and extracts.

### 1.2.2 CICS Performance Monitor

CICS PM is an online monitoring tool that provides real-time performance management, monitoring, and troubleshooting solutions for CICS Transaction Server (TS). It allows you to detect performance problems early, identify the cause, and change system and resource parameters to avoid problems. CICS PM uses the CICSPlex System Manager (SM) Web User Interface (WUI) server component of CICS TS.

CICS PM complements CICS PA for online analysis. The product is based on a standard application programming interface (API). It is built on facilities of CICSPlex SM, which is an integral part of CICS TS. The underlying CICSPlex SM infrastructure is transparent to the CICS PM user, so little knowledge of CICSPlex SM is necessary to operate CICS PM.

CICS PM consists of two components:

- **CICS PM workstation client**: You download this component to a Windows® workstation. It provides a graphical user interface (GUI) that displays data provided by a supporting application of CICS PM installed in CICS TS. The client GUI consists of three components:
  - **Threshold definitional component**: Allows the user to create and maintain threshold definitions.
  - **Monitoring component**: Provides the ability to monitor the status of multiple CICS regions. When a threshold is triggered, an event is created. Information about multiple events is aggregated into an event view for rapid problem identification. Navigation to the CICS PM view sets facilitates problem resolution by providing more detailed information.
  - **History definitional and reporting component**: Allows the user to create and maintain history definitions for completed task history. The status of installed history definitions can also be monitored. The reporting component launches Web-based views of completed task history data.

- **CICS PM server**: This component provides a comprehensive series of view sets specifically tailored for performance analysis and problem determination. These view sets provide real-time access to all CICS systems and resource-related performance data. They also provide access to all the task-level performance data collected by the CMF.

The CICS system-level view sets include:

- CICS region
- CICS storage (dynamic storage area (DSA) and subpool usage)
- Transaction manager
- CICS dispatcher
- Loader
- Temporary storage
- Transient data
- DB2 connection
- Recovery manager, including unit of work (UOW) analysis
- Enqueue pools
- JVM pool
- Global and task-related user exits

The view sets provide access to all resources defined to the CICS systems, including:
- Transactions
- Transaction classes
- Programs
- Files, data tables, and local shared resource (LSR) pools
- Connections
- Log streams
- Journals
- Temporary storage queues
- Transient data queues
- Terminals
- DB2 entries
- System and transaction dumps
- Enterprise Java resources

Other view sets are provided, giving access to the performance information about all the active and optionally recently completed tasks in the CICS systems.

1.2.3 Benefits of CICS Performance Analyzer and CICS Performance Monitor

This section summarizes the benefits provided by both CICS performance tools.

The benefits provided by CICS PA are:
- Ease of use (no additional setup or customization required)
- Easily customizable performance reports
- Detailed and summary reports on all aspects of CICS system activity and resource usage
- Extracts for graphing and analysis by external programs, such as DB2 or PC tools (for example, Lotus® 1-2-3®)
- Historical Database repository for CMF performance class data
- CICS PA can help to:
  - Analyze CICS application performance
  - Improve CICS resource usage
  - Evaluate the effects of CICS system tuning efforts
  - Improve transaction response time
  - Provide ongoing system management and measurement reports
  - Increase availability of resources
  - Increase the productivity of system and application programmers
  - Provide awareness of usage trends, assisting with future growth estimates

The benefits provided by CICS PM are:
- Improved CICS systems availability
- Reduced system management costs
- Ability to detect performance problems early
- Enablement of changes to key system and resource parameters
- Enablement of online interactive access to performance data of recently completed tasks
- Easy-to-use comprehensive interface for exception management
- Built on standard APIs and proven CICS TS system management technologies
- Ease of installation and setup
The two products, CICS PA and CICS PM, complement each other. Table 1-1 provides a functional comparison between the two products.

**Table 1-1  CICS PA and CICS PM comparison**

<table>
<thead>
<tr>
<th>Function</th>
<th>CICS PA</th>
<th>CICS PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invocation</td>
<td>Offline</td>
<td>Online</td>
</tr>
<tr>
<td>Mode</td>
<td>Passive</td>
<td>Active</td>
</tr>
<tr>
<td>Input origin/output destination</td>
<td>3270/reports</td>
<td>Windows workstation</td>
</tr>
<tr>
<td>Output format</td>
<td>Tabular, graphical, extracts</td>
<td>GUI</td>
</tr>
<tr>
<td>Information source</td>
<td>SMF</td>
<td>CICSPlex SM</td>
</tr>
<tr>
<td>History</td>
<td>As far back as you keep SMF data, or maintained in CICS PA HDB</td>
<td>Recent</td>
</tr>
<tr>
<td>Deals with</td>
<td>Exceptions, performance, transaction resource usage history</td>
<td>Alerts</td>
</tr>
<tr>
<td>Additional usage</td>
<td>Capacity planning</td>
<td>Status monitoring</td>
</tr>
</tbody>
</table>

**1.2.4 CICS Online Transmission Time Optimizer**

CICS Online Transmission Time Optimizer for z/OS (CICS OTTO) is a run-time tool that optimizes:

- Data streams directed to 3270-type display stations, printers, or both
- Data streams directed to SCS-type printers
- Data streams directed to banking terminals 3600/4700

The supported CICS releases are:

- CICS Transaction Server for z/OS, Version 2.1 and 2.2
- CICS Transaction Server for OS/390, Version 1

**1.2.5 CICS Interdependency Analyzer**

CICS Interdependency Analyzer for z/OS and OS/390 (CICS IA) is a run-time tool that:

- Analyzes resource interdependencies
  - What a CICS region has in it
  - What resources a transaction needs to run
  - Which programs use which resources
  - What resources are no longer used
- Writes report data to a DB2 database

The supported CICS releases are:

- CICS Transaction Server for z/OS, Version 2.1 and 2.2
- CICS Transaction Server for OS/390, Version 1

**1.2.6 CICS Business Event Publisher for MQSeries**

CICS Business Event Publisher for MQSeries (CBEP) enables a rapid extension of existing applications running in CICS Transaction Server V1.3 or CICS Transaction Server V2.2. CBEP generates user-defined MQSeries messages as a side effect when certain EXEC CICS commands are executed by a CICS application. Message generation is transparent to
the application program. CICS Business Event Publisher for MQSeries supports the following functions:

- Enables customizable MQSeries messages and queues based on rules
- Provides real-time data propagation
- Offers external logging or notification of CICS-related activity
- Enables non-CICS functions to act as write-only data repositories

### 1.2.7 IBM Session Manager for z/OS

IBM Session Manager for z/OS provides IBM Virtual Telecommunications Access Method (VTAM®) and Transmission Control Protocol/Internet Protocol (TCP/IP) users a secure and user-friendly way to access multiple IBM OS/390 and IBM z/OS systems from a single 3270 terminal. With a highly secure, single sign-on capability, users can access all your business applications from multiple concurrent, virtual sessions. Session Manager supports the following functions:

- Enables a common user interface for all TCP/IP and VTAM applications
- Eliminates redundant and time-consuming logon and logoff activities and application switching
- Uses a single network connection to establish multiple concurrent sessions
- Allows you to easily and efficiently manage multiple sessions and different types of user groups

Session Manager for z/OS can help you:

- Reduce training costs
  Point-and-click ease means users don’t have to learn an entirely new skill set.
- Enhance system usage
- Increase security
- Reduce the cost and effort associated with network administration
- Provide access to mainframe applications from distributed or workstation programs

### 1.2.8 CICS VSAM Recovery

CICS VSAM Recovery (CICSVR) recovers lost or damaged Virtual Storage Access Method (VSAM) data. CICSVR is for organizations where the availability and integrity of VSAM data is vital. CICSVR provides:

- A screen interface to help assess the situation and initiate forward recovery.
- Automatic backups and log streams and log stream copies required for recovery.
- Forward recovery to recover lost or damaged VSAM data sets.
- Multiple data set recovery in a single run.
- An ISPF screen interface that complies with Common User Access (CUA)
  The interface can be used to direct CICSVR to create and submit a job to restore VSAM data sets from a logical backup and perform a forward recovery.
- Automatic restore of VSAM data sets from logical backups created by DFSMShsm and DFSMSdss.
- Support of backup-while-open (BWO) data sets
  This enables you to create BWO backups when a data set is open and being updated by CICS. CICSVR can restore and recover VSAM data sets from backups created by the BWO facility.
1.3 Data used by CICS Performance Analyzer

This section discusses the types of SMF data that CICS Performance Analyzer can process.

1.3.1 CICS Monitoring Facility data (SMF 110 records)

CICS monitoring collects data about the performance of all user- and CICS-supplied transactions during online processing for later offline analysis. The records produced by CICS monitoring are MVS System Management Facility (SMF) type 110 records. They are written to an SMF data set.

The CMF enables you to collect the following types or classes of monitoring data:

- Performance class data
- Exception class data
- Transaction resource class data

Controlling CICS monitoring

When you start CICS, you switch on the monitoring facility by specifying the system initialization parameter MN=ON. MN=OFF is the default setting. You can select the classes of monitoring data that you want to be collected using the MNPER, MNEXC, and MNRES system initialization parameters. You can request the collection of any combination of performance class data, exception class data, and transaction resource class data. You can change the class settings whether the monitoring facility is ON or OFF. For details about all the system initialization parameters that control monitoring activities, see the CICS Transaction Server for z/OS CICS System Definition Guide, SC34-5988.

When CICS is running, you can control the monitoring facility dynamically. As with CICS initialization, you can switch monitoring on or off. You can also change the classes of monitoring data that are being collected. There are two ways to do this:

- Use the master terminal CEMT INQ|SET MONITOR command, which is described in CICS Transaction Server for z/OS CICS Supplied Transactions, SC34-5992.
- Use the EXEC CICS INQUIRE MONITOR and SET MONITOR commands (see the CICS Transaction Server for z/OS CICS System Programming Reference, SC34-5995).

If you activate a class of monitoring data in the middle of a run, the data for that class becomes available only for transactions that are started thereafter. You cannot change the classes of monitoring data that is collected for a transaction after it has started. It is often preferable, particularly for long-running transactions, to start all classes of monitoring data at CICS initialization.

How CICS monitoring data is passed to SMF

The various CICS monitoring class records are not written to SMF in the same way as explained here.

Performance data records are written to a performance record buffer, which is defined and controlled by CICS as the records are produced. The performance records are passed to SMF for processing:

- When the buffer is full
- When the performance class of monitoring is switched off
- When CICS itself quiesces

When monitoring itself is deactivated or when there is an immediate shutdown of CICS, the performance records are not written to SMF and the data is lost.
Exception records are passed directly to SMF when the exception condition completes. Each exception record describes one exception condition. Performance and exception records can be matched by transaction number (TRANNUM) or network unit-of-work ID (NETUOWPX and NETUOWSX).

Transaction resource data records are written to a transaction resource record buffer, which is defined and controlled by CICS, as the records are produced. The transaction resource records are passed to SMF for processing:

- When the buffer is full
- When the transaction resource class of monitoring is switched off
- When CICS itself becomes quiescent

When monitoring itself is deactivated or when there is an immediate shutdown of CICS, the transaction resource records are not written to SMF and the data is lost.

Performance class data
Performance class data is detailed transaction-level information, such as the processor and elapsed time for a transaction, or the time spent waiting for input/output (I/O). At least one performance record is written for each transaction that is being monitored.

Performance class data provides detailed, resource-level data that can be used for accounting, performance analysis, and capacity planning. This data contains information relating to individual task resource usage. It is completed for each task when the task terminates.

You can enable performance class monitoring by coding MNPER=ON (together with MN=ON) as a system initialization parameter. Alternatively, you can use one of the following two commands to enable performance class monitoring dynamically:

CEMT SET MONITOR ON PERF
EXEC CICS SET MONITOR STATUS(ON) PERFCLASS(PERF)

You can use this information periodically to calculate the charges applicable to different tasks. If you want to set up algorithms for charging users for resources used by them, you can use this class of data collection to update the charging information in your organization’s accounting programs. For older versions of CICS, charging primarily on exact resource usage was not recommended, because of the overhead involved in obtaining these figures.

Exception class data
Exception class monitoring data is information about CICS resource shortages suffered by a transaction. This data highlights possible problems in CICS system operation. It is intended to help you identify system constraints that affect the performance of your transactions. There is one exception record for each type of exception condition. The exception records are produced and written to SMF as soon as the resource shortage encountered by the transaction is resolved. Exception records are produced for each of the following resource shortages:

- Wait for storage in the CDSA
- Wait for storage in the UDSA
- Wait for storage in the SDSA
- Wait for storage in the RDSA
- Wait for storage in the ECDSA
- Wait for storage in the EUDSA
- Wait for storage in the ESDSA
- Wait for storage in the ERDSA
- Wait for auxiliary temporary storage
- Wait for auxiliary temporary storage string
- Wait for auxiliary temporary storage buffer
- Wait for coupling facility data tables locking (request) slot
- Wait for coupling facility data tables non-locking (request) slot
- Wait for file buffer
- Wait for LSRPOOL string
- Wait for file string

If the monitoring performance class is also recorded, the performance class record for the transaction includes the total elapsed time that the transaction was delayed by a CICS system resource shortage. This is measured by the exception class and the number of exceptions encountered by the transaction. The exception class records can be linked to the performance class records either by the transaction sequence number or by the network unit-of-work ID.

You can enable exception class monitoring by specifying the MNEXC=ON (together with MN=ON) system initialization parameter. Alternatively, you can use one of the following two commands to enable exception class monitoring dynamically:

CEMT SET MONITOR ON EXCEPT
EXEC CICS SET MONITOR STATUS(ON) EXCEPTCLASS(EXCEPT)

Transaction resource class data
Transaction resource class data is a new CICS TS monitoring feature introduced by authorized program analysis report (APAR). Ensure that you apply the relevant program temporary fixes (PTFs) in Table 2-1 on page 26.

Transaction resource class data provides additional transaction-level information about individual resources accessed by a transaction. Currently, the transaction resource class covers file and temporary storage resources only. The maximum number of files and temporary storage queues monitored for each transaction is limited by the FILE and TSQUEUE parameters on the DFHMCT TYPE=INITIAL macro. The default is FILE=8 for files and TSQUEUE=4 for temporary storage queues. Therefore, you may need to assemble an monitoring control table (MCT) that specifies the FILE option, TSQUEUE option, or both options if the default values are insufficient, or if you do not want to collect transaction resource data for either files or temporary storage queues. One transaction resource record is written for each transaction that is being monitored. This happens provided that the transaction accesses at least one of the resources for which monitoring data is requested.

Performance class data also provides information about file and temporary storage queue accesses. However, this information in the performance record is given in total only for all files and all temporary storage queues. Transaction resource class data breaks down this information by individual file name and temporary storage queue name, up to the maximum number specified in the MCT. Transaction resource information is completed for each task when the task terminates.

You enable transaction resource class monitoring at startup by coding MNRES=ON (together with MN=ON) as a system initialization parameter. Alternatively, you can use one of the following two commands to enable transaction resource class monitoring dynamically:

CEMT SET MONITOR ON RESRCE
EXEC CICS SET MONITOR STATUS(ON) RESRCECLASS(RESRCE)

Event monitoring points
CICS monitoring data is collected at system-defined event monitoring points (EMPs) in the CICS code. Although you cannot relocate these monitoring points, you can choose which
classes of monitoring data you want to collect. For programming information about CICS monitoring, see CICS Transaction Server for z/OS CICS Customization Guide, SC34-5989.

If you want to gather more performance class data than is provided at the system-defined EMPs, you can code additional EMPs in your application programs. At these points, you can add or change up to 16384 bytes of user data in each performance record. Within this limit you can have, for each ENTRYNAME qualifier, any combination of:

- Between 0 and 256 counters
- Between 0 and 256 clocks
- A single 8192-byte character string

You can use these additional EMPs to count the number of times a certain event occurs, or to time the interval between two events. If the performance class was active when a transaction was started, but was not active when a user EMP was issued, the operations defined in that user EMP still execute on that transaction’s monitoring area. The DELIVER option results in a loss of data at this point, because the generated performance record cannot be output while the performance class is not active. If the performance class is not active when a transaction was started, the user EMP has no effect.

User EMPs are used in combination with the EXEC CICS MONITOR command. This command activates and deactivates them. For programming information about this command, refer to the CICS Transaction Server for z/OS CICS Application Programming Reference, SC34-5994.

Additional EMPs are provided in some IBM program products, such as database control (DBCTL). From the CICS point of view, these are like any other user-defined EMP. EMPs in user applications and in IBM program products are identified by a decimal number. The numbers 1 through 199 are available for EMPs in user applications. The numbers from 200 through 255 are for use in IBM program products. The numbers can be qualified with an entry name, so that you can use each number more than once. For example, PROGA.1, PROGB.1, and PROGC.1 identify three different EMPs because they have different entry names.

For each user-defined EMP, there must be a corresponding MCT entry, which has the same identification number and entry name as the EMP that it describes. You do not have to assign entry names and numbers to system-defined EMPs. Nor do you have to code MCT entries for them.

Here are some ideas for using the CICS and user fields provided with the CICS Monitoring Facility:

- If you want to time how long it takes to perform a table lookup routine within an application, code an EMP with, for instance ID=50, just before the table lookup routine and an EMP with ID=51 just after the routine. The system programmer codes a TYPE=EMP operand in the MCT for ID=50 to start user clock 1. You also code a TYPE=EMP operand for ID=51 to stop user clock 1. The application executes. When EMP 50 is processed, user clock 1 is started. When EMP 51 is processed, the clock is stopped.

- You can use one user field to accumulate an installation accounting unit. For example, you may count different amounts for different types of transaction. Or, in a browsing application, you may count one unit for each record scanned and not selected, and three for each record selected.

- You can also treat the full word count fields as 32-bit flag fields to indicate special situations, for example, out-of-line situations in the applications, operator errors, and so on. CICS includes facilities to turn individual bits or groups of bits on or off in these counts.

- You can use the performance clocks to accumulate the time taken for I/O, DL/I scheduling, and so on. It usually includes any waiting for the transaction to regain control after the
requested operation has completed. Because the periods are counted as well as added, you can get the average time waiting for I/O as well as the total. If you want to highlight an unusually long individual case, set a flag on in a user count as explained earlier.

- A use of the performance character string is for systems in which one transaction ID is used for widely differing functions. The application can enter a subsidiary ID into the string to indicate which particular variant of the transaction applies in each case. This use of user EMPs is now catered for by the Application Naming function.

Some users have a single transaction ID so that all user input is routed through a common prologue program for security checking, for example. In this case, it is easy to record the subtransaction identifier during this prologue. However, it is equally possible to route transactions with different identifiers to the same program, in which case this technique is not necessary.

**Monitoring control table**

You use the monitoring control table for the following reasons:

- To specify the type of resource for which you want to collect Transaction Resource Monitoring data (DFHMCT TYPE=INITIAL) and the maximum number of files (FILE= option) and temporary storage queues (TSQUEUE= option) for Transaction Resource Monitoring
- To enable Application Naming support, which makes available the CICS-generated DFHAPPL EMPs to your application programs (DFHMCT TYPE=INITIAL)
- To notify CICS about the EMPs that you coded in your application programs and about the data that is to be collected at these points (DFHMCT TYPE=EMP)
- To notify CICS that you do not want certain system-defined performance data to be recorded during a particular CICS run (DFHMCT TYPE=RECORD)

IMS DBCTL users can collect DBCTL statistics in the CMF performance class records by including the DFHSMCTD copy member in the MCT definition.

You can find full details about the MCT in the CICS Transaction Server for z/OS CICS Resource Definition Guide, SC34-5990. Examples of MCT coding are included with the programming information in the CICS Transaction Server for z/OS CICS Customization Guide, SC34-5989.

Four sample monitoring control tables are also provided in CICSTS22.CICS.SDFHSAMP:

- **DFHMCTT$**: For terminal-owning regions (TORs)
- **DFHMCTA$**: For application-owning regions (AORs)
- **DFHMCTD$**: For application-owning regions (AORs) with DBCTL
- **DFHMCTFS$**: For file-owning regions (FORs)

These samples show how to use the EXCLUDE and INCLUDE operands to reduce the size of the performance class record and reduce the volume of data written by CICS to SMF.

### 1.3.2 DB2 accounting data (SMF 101 records)

DB2 accounting data is written as SMF type 101 records.

**DB2 accounting trace**

The DB2 accounting trace provides information related to application programs, including:

- Start and stop times
- Number of commits and aborts
The number of times certain SQL statements are issued
Number of buffer pool requests
Counts of certain locking events
Processor resources consumed
Thread wait times for various events
RID pool processing
Distributed processing
Resource limit facility statistics

The DB2 trace begins collecting this data at successful thread allocation to DB2. It writes a completed record when the thread terminates or when the authorization ID changes.

DB2 accounting records are produced when a thread is terminated or sign-on occurs. This means that the period reported in the DB2 accounting record is the time between start or user sign-on (if reusing a thread previously used by another user) and thread termination or another sign-on. You can use the ACCOUNTREC(TXID) parameter in the DB2ENTRY or DB2CONN to cause a DB2 accounting record to be produced when the transaction ID changes, and when the thread terminates or another sign-on occurs.

For thread reuse, this means that many users are included in the same record, which can cause difficulties for both accounting and problem determination. The ACCOUNTREC(TASK) or ACCOUNTREC(UOW) settings in a DB2ENTRY or DB2CONN provide more granularity. This is because a record is produced for each user. It involves the passing of a token between CICS and DB2, which is present in both CICS and DB2 traces.

ACCOUNTREC(TASK) ensures that there is a minimum of one accounting record for each task. There can be more depending on thread reuse.

For more information about accounting and monitoring in a CICS DB2 environment, refer to the CICS Transaction Server for z/OS CICS DB2 Guide, SC34-6014. For more information about setting up DB2 accounting, refer to the DB2 UDB for OS/390 and z/OS Administration Guide, SC26-9931.

**Accounting CLASS 1 processor time**
For accounting CLASS 1, a task processor timer is created when the task control block (TCB) is attached. When a thread to DB2 starts, the timer value is saved. When the thread is terminated (or the authorization ID is changed), then the timer is checked again. Both the timer start and end values are recorded in the SMF 101 records (the DB2 accounting record).

**Accounting CLASS 2 processor time**
For accounting CLASS 2, the timer is checked on every entry and exit from DB2 to record the “IN DB2” time in the SMF type 101 record. In this case, it is the difference that is stored in the record.

### 1.3.3 WebSphere MQ accounting data (SMF 116 records)

WebSphere MQ accounting data is written as SMF type 116 records.

**Accounting class 1 and class 3**
WebSphere MQ accounting information can be collected for three subtypes:

0 Message manager accounting records (how much of the central processing unit (CPU) was spent processing WebSphere MQ API calls and the number of MQPUT and MQGET calls)
This information is produced when a named task disconnects from WebSphere MQ. The information contained within the record may cover many hours.

1 Accounting data for each task, at thread and queue level
2 Additional queue-level accounting data (if the task uses more queues than can fit in the subtype 1 record)

Subtype 0 is produced with trace class 1. Subtypes 1 and 2 are produced with trace class 3.

**MQ accounting trace**

You can start the WebSphere MQ trace facility at any time by issuing the WebSphere MQ START TRACE command.

Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts. It must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some MQ accounting data automatically if you specify YES in the SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro.

You cannot use this method to start collecting class 3 accounting information (thread-level and queue-level accounting). You must use the START TRACE command to do this. However, you can include the command in your CSQINP2 input data set so that the trace is started automatically when you start your queue manager.

For more information about setting up WebSphere MQ accounting, refer to the *WebSphere MQ for z/OS System Setup Guide*, SC34-6052.

### 1.3.4 MVS System Logger data (SMF 88 records)

System Logger produces SMF record type 88 to record the System Logger activity of a single system in a sysplex. These records are written to the active SMF data set on the system.

**Capacity planning**

For capacity planning purposes, we recommend that you view the steady-state performance requirements of an application. Various flags in the SMF record type 88 highlight exception scenarios for additional analysis or changes in report processing.

**Record type 88**

Record type 88 focuses on the logstream data for a system in a sysplex, including use of *interim storage*. Interim storage is where log data is initially written, before being written to direct access storage device (DASD) log data sets. You can quickly access data in interim storage without incurring DASD I/O. In a coupling facility log stream, interim storage for log data is in coupling facility list structures. In a DASD-only log stream, interim storage for log data is contained in local storage buffers on the system and duplexed to staging data sets. Using record type 88 can help an installation avoid the STRUCTURE FULL exception, and perform other tuning, capacity planning analysis, or both.

Given a specific log stream, a record type 88 summarizes all of that log stream's activity on that system, as long as at least one address space is connected to the log stream on that system. If no System Logger write activity is performed on the log stream during a particular
SMF interval, a record is produced showing zero for the various System Logger activity total fields.

The System Logger SMF record is cut for all log streams connected at the expiration of the SMF global recording interval. Record type 88 is also triggered by the disconnection of the last log stream on that system.

SMF fields relating to resource events, either structure full or staging data set full conditions, should be handled depending on:

- Whether the resource is shared sysplex-wide and each system will take action
- Whether the resource is shared sysplex-wide but only one system will take action
- Whether the resource is consumed on a system-local basis

To obtain a sysplex-wide view of System Logger activity, correct processing for most SMF 88 data fields is to sum the field contents for the target interval across all the SMF 88 records produced in the sysplex. There are, however, exceptions to this rule. Because each system must take its own action — that is, wait for an ENF signal indicating that System Logger is available — an analysis program should use the maximum value for these fields: SMF88ERI, SMF88ERC, and SMF88ESF. For example, if a structure rebuild is initiated in a sysplex with three systems, the event is recorded on all three systems. The correct number of structure rebuild initiations is not three, but one or the maximum number provided SMF88ERI.

For DASD-only log streams, staging data sets are a required part of the logstream configuration. For coupling facility log streams, use of staging data sets implies a trade-off between performance workload and data integrity. You should try to tune the staging data set size to minimize the number of Staging_Dataset_Threshold_Hit conditions. Without this type of tuning, such conditions can impact performance during staging data set processing. Only an installation can determine what the proper trade-off between performance and data integrity should be.

Because System Logger maintains interim storage differently for coupling facility based log stream versus DASD-only log streams, the difference is reflected in the SMF record 88 report:

- For a coupling facility based log stream, the Structure (Interim Storage) section of the record 88 report shows information about the usage of coupling facility structure space allocated for a log stream and the flow of log data through the structure.
- For a DASD-only log stream, the Structure (Interim Storage) section of the record 88 report shows information about usage of staging dataset space and the flow of data through the staging data set for the log stream.

Not all fields in the Structure (Interim Storage) section of the record 88 report apply to DASD-only log streams. For a DASD-only log stream, fields that do not apply contain zeros. The SMF88STN field contains *DASDONLY* for a DASD-only log stream because there is no structure name.

### 1.4 Other relevant CICS data and utilities

Other CICS-provided tools can help you gather CICS performance data. They are not required by CICS PA, but they may assist your analysis and decision-making when using CICS PA and interpreting the output.

CICS provides two statistics utilities programs and two programs for processing CICS monitoring data written to SMF. In addition, you can use the DFHJUP utility to copy data from system SMF data sets.
1.4.1 CICS statistics

CICS management modules control how events are managed by CICS. As events occur, CICS produces information that is available to you as system and resource statistics. The resources controlled by CICS include files, databases, journals, transactions, programs, and tasks. Resources that CICS manages, and values that CICS uses in its record-keeping role, are defined in one of the following ways:

- Online by the CICS CEDA transaction
- Offline by the CICS system definition (CSD) utility program DFHCSDUP
- Offline by CICS control table macros

Statistics are collected during CICS online processing for later offline analysis. The statistics domain writes statistics records to an SMF data set. The records are of SMF type 110, sub-type 002. Monitoring records and some journaling records are also written to the SMF data set as type 110 records. For programming information about SMF, DFHCSDUP, and about other SMF data set considerations, see the CICS Transaction Server for z/OS CICS Customization Guide, SC34-5989.

Types of statistics data

CICS produces five types of statistics:

- **Interval statistics**: These are gathered by CICS during a specified interval. You can change the interval value using the STATINT system initialization parameter, using CEMT SET STATISTICS, or using the EXEC CICS SET STATISTICS command.

- **End-of-day statistics**: These statistics are gathered on three occasions:
  - At the end-of-day expiry time
  - When CICS becomes quiescent (normal shutdown)
  - When CICS terminates (immediate shutdown)

  The end-of-day value defines a logical point in the 24-hour operation of CICS. You can change the end-of-day value using the STATEOD system initialization parameter, using CEMT SET STATISTICS, or using the EXEC CICS SET STATISTICS command.

- **Requested statistics**: These are statistics that the user requested by using one of the following three commands:

  CEMT PERFORM STATISTICS RECORD
  EXEC CICS PERFORM STATISTICS RECORD
  EXEC CICS SET STATISTICS ON|OFF RECORDNOW

  These commands cause the statistics to be written to the SMF data set immediately, instead of waiting for the current interval to expire. For more details about CEMT commands, see CICS Transaction Server for z/OS CICS Supplied Transactions, SC34-5992. For programming information about the equivalent EXEC CICS commands, see the CICS Transaction Server for z/OS CICS System Programming Reference, SC34-5995.

- **Requested reset statistics**: These statistics differ from requested statistics in that all statistics are collected and statistics counters are reset. You can reset the statistics counters using the CEMT or EXEC CICS PERFORM/SET commands.

- **Unsolicited statistics**: These statistics are automatically gathered by CICS for dynamically allocated and deallocated resources. CICS writes these statistics to SMF just before the resource is deleted regardless of the status of statistics recording.
Processing CICS statistics
You may find it particularly useful to process the statistics records and the monitoring records together. This is because statistics provide resource and system information that is complementary to the transaction data produced by CICS monitoring.

There are several ways to process CICS statistics, including:

- Using the CICS DFHSTUP offline utility: For guidance about retrieving CICS statistics from SMF, and about running DFHSTUP, see the CICS Transaction Server for z/OS CICS Operations and Utilities Guide, SC34-5991.

- Writing your own program to report and analyze the statistics: For details about the statistics record types, see the assembler DSECTs named in each set of statistics. For programming information about the formats of CICS statistics SMF records, see the CICS Transaction Server for z/OS CICS Customization Guide, SC34-5989.

- Using the sample statistics program (DFH0STAT): You can use the statistics sample program, DFH0STAT, to produce online reports from the CICS statistics data. The program demonstrates the use of the EXEC CICS INQUIRE and EXEC CICS COLLECT STATISTICS commands to produce an analysis of a CICS system. You can use the sample program as provided or modify it to suit your needs.

- Using Tivoli® Decision Support to process CICS SMF records to produce joint reports with data from other SMF records.

1.4.2 The sample statistics program: DFH0STAT
The sample statistics program, DFH0STAT, produces a report that shows comprehensive system information about CICS resources. It also shows an overview of the MVS storage in use. The program demonstrates how you can use EXEC CICS INQUIRE and EXEC CICS COLLECT STATISTICS commands to produce an analysis of your CICS regions. You can use the sample program as supplied, or modify it to suit your needs.

DFH0STAT does not report on terminals, DBCTL resources, front-end programming interface (FEPI) resources, dumps, the table manager, and the user domain. If you require statistical information about these areas, you can obtain it using DFHSTUP, the statistics utility program.

Keep in mind that DFH0STAT does not always report to the maximum capacity of certain large statistics fields. If your CICS system is unusually large or very busy, and you have a long statistics interval, check that the statistics values have not overflowed. To avoid this problem, reduce the length of your statistics interval, or use DFHSTUP.

1.4.3 Statistics utility program: DFHSTUP
The statistics utility program, DFHSTUP, prepares and prints reports offline, using the CICS statistics data recorded on the MVS system management facilities (SMF) SYS1.MANx data sets. To enable the CICS statistics domain to record interval statistics on these SMF data sets, you must specify the STATRCD=ON system initialization parameter. The other statistics record types (unsolicited, requested and end-of-day) are written regardless of the setting of the STATRCD option.

For information about the SMF data sets, see the OS/390 MVS System Management Facilities (SMF), GC28-1783. For information about what CICS data is recorded on the SMF data sets, and about interpreting CICS statistics output in the DFHSTUP report, see the CICS Transaction Server for z/OS Performance Guide, SC34-6009. For a description of the STATRCD system initialization parameter, see the CICS Transaction Server for z/OS CICS System Definition Guide, SC34-5988.
Use the version of the DFHSTUP program from the same release of CICS as the data that it is to process.

For more information about DFHSTUP, refer to *CICS Transaction Server for z/OS CICS Operations and Utilities Guide*, SC34-5991.

### 1.4.4 Monitoring dictionary utility program: DFHMNDUP

DFHMNDUP is a utility program that generates a performance dictionary record, in a sequential data set, for use with monitoring data extracted from SMF data sets. When CICS monitoring is switched on, and you activate the monitoring performance class (MNPER=ON), CICS first writes a performance dictionary record to the current SMF data set. Then it begins to write the monitoring performance data records.

A new dictionary record, which always precedes the monitoring data it relates to, is written whenever you start CICS with the performance class active and CICS monitoring turned on. This record is also written when you change the status of the monitoring performance class from inactive to active, with CICS monitoring turned on. If monitoring is turned off and the monitoring performance class is switched from inactive to active, a dictionary record is scheduled from the next time monitoring is activated.

Any monitoring utility program that processes performance data must read the dictionary record that relates to the data being processed before it attempts to analyze the data. However, if SMF switches data sets during the period when CICS monitoring is writing performance data, CICS does not write a new dictionary record. Therefore a CICS performance dictionary record is not the first monitoring performance record on the new SMF data set. The DFHMNDUP program provides a solution to the problem posed by SMF data sets that do not contain a dictionary record.

The CICS PA System Definitions facility uses DFHMNDUP to create a Dictionary record on request.

### 1.4.5 Sample monitoring data print program: DFH$MOLS

DFH$MOLS is a print program for CICS monitoring data. It is a sample program that you can modify or adapt to your own purposes. It is intended to show how you can code your own monitoring utility program to print CICS monitoring data.

The job tasks that are involved to process CICS monitoring data are:

1. Unload the SMF data set or sets so that the SMF data is available for processing by a CICS utility. For information about unloading SMF data sets, refer to *OS/390 MVS System Management Facilities (SMF)*, GC28-1783.

2. Run the DFH$MOLS program to print monitoring records, which you can optionally select and sort by means of control statements.

The DFH$MOLS program is a data reduction program that is designed to produce reports from the data collected by the CICS monitoring domain (MN), and written to SMF data sets.

The CICS Transaction Server for z/OS, Version 2 Release 2, DFH$MOLS can process SMF 110 monitoring data records for earlier CICS Transaction Server versions and releases. However, DFH$MOLS cannot process monitoring data written by a release of CICS later than itself. Therefore, you should *always* use the DFH$MOLS from the highest version or release available to you.
You run the DFH$MOLS program in a batch region to process any CICS SMF type 110 monitoring records that are present in an unloaded SMF data set. You can write the data set to either a temporary or cataloged data set. You can determine the scope of the report or reports by supplying control statements in the SYSIN data set.

The program reads, formats, and prints the CICS monitoring data, which is packaged in the format:

```plaintext
[SMF HEADER .][SMF PRODUCT SECTION .][CICS DATA SECTION ]
```

The CICS data section in a monitoring record is one of the following types:

- A dictionary data section, consisting of a sequence of dictionary entries
- A performance data section, consisting of a sequence of field connectors followed by one or more performance records (monitoring record type 3)
- An exception data section, consisting of one exception record (monitoring record type 4)
- A transaction resource data section, consisting of one or more Transaction Resource Monitoring records (monitoring record type 5)

For programming information about the structure of CICS SMF type 110, and how the monitoring data is packaged within the SMF records, see the *CICS Transaction Server for z/OS CICS Customization Guide*, SC34-5989. The DFH$MOLS program reads the SMF data and formats and prints it. If you want to analyze the data using your own routines, this is the point at which you can link to a user-written analysis program.

The DFH$MOLS program prints about one page per task. Therefore, be sure to specify only those items that you need using the DFH$MOLS program control statements.

Note that the DFH$MOLS program requires a performance dictionary record to process monitoring performance data. When it locates a dictionary record, it builds an in-store dictionary and processes any subsequent performance data using this dictionary. Whenever it reads a new dictionary record, the current dictionary is released and a new in-store dictionary is built. The dictionary record must appear before any related performance data. Otherwise the DFH$MOLS program abends. Note that monitoring exception records does not require a dictionary, so they can precede the first dictionary record and still be successfully processed.

### 1.4.6 Journal utility program: DFHJUP

In general, this batch utility is used to read, process, copy, or print CICS log data in MVS System Logger log streams and SMF data sets. CICS can write user journal and autojournal data to SMF data sets rather than to log streams. It is useful to copy and print this data.

### 1.5 Other relevant information sources

There are several tools for obtaining system performance data relevant to evaluating performance of the CICS system.

#### 1.5.1 System Management Facility

SMF collects and records system and job-related information that you can use in:

- Billing users
- Reporting reliability
- Analyzing your configuration
CICS PA processes the following SMF record types:
- CICS Monitoring Facility (type 110)
- DB2 accounting (type 101)
- WebSphere MQ accounting (type 116)
- System Logger (type 88)

1.5.2 Resource Management Facility

Resource Management Facility collects system-wide data that describes the processor activity (WAIT time), I/O activity (channel and device usage), main storage activity (demand and swap paging statistics), and system resources manager (SRM) activity (workload).

RMF™ is a centralized measurement tool that monitors system activity to collect performance and capacity planning data. The analysis of RMF data provides the basis for tuning the system to user requirements. You can also use it to track resource usage.

You can also use RMF workload activity reports in conjunction with the CICS PA Workload Activity report. This combination helps you to understand from a CICS perspective how well your CICS transactions are meeting their response time goals.

1.5.3 Generalized Trace Facility

CICS trace entries can be recorded through Generalized Trace Facility (GTF), and reports produced with Interactive Program Control System (IPCS). GTF is an integral part of the z/OS system and traces the events of DASD seeking addresses on start I/O instructions, SRM activity, page faults, I/O activity, and supervisor services. Execution options specify the system events to be traced.

CICS GTF data can be combined with data for other components, for example, the data about use of VTAM buffers.

GTF is generally used to monitor short periods of system activity. You should run it accordingly. No data reduction programs are provided with GTF. To extract and summarize the data into a meaningful and manageable form, you can either write a data reduction program yourself or use one of the program offerings that are available.

1.5.4 Tivoli Decision Support for OS/390

This product collects and analyzes data from CICS and other IBM systems and products. With Tivoli Decision Support, you can build reports that help you manage:
- Service levels
- Availability
- Performance and tuning
- Capacity planning
- Change and problem management
- Accounting

Several ready-made reports are available. In addition, you can generate your own reports to meet specific needs.
A quick start to CICS Performance Analyzer

This chapter introduces you to using the CICS Performance Analyzer (PA) screen. It also helps you to understand the CICS PA concepts. Follow along if this is your first time using CICS PA.

We start with an overview of CICS PA operation, the system requirements, our Interactive System Productivity Facility (ISPF) setup, and preparing the system management facility (SMF) data. Then we show you how to:

- Start CICS PA
- Use the Take-up facility to easily define CICS systems and SMF data files to CICS PA
- Maintain system definitions
- Define Report Sets to request reports and extracts from our defined systems and files
- Submit report requests to run in batch
- View report output
- Tailor reports using Report Forms
- Filter the data using selection criteria and Object Lists

This chapter introduces you to only a fraction of the CICS PA functionality. However, the other reports, extracts, and functions offered by CICS PA are essentially variations or extensions of what is covered here.

The Historical Database (HDB) facility employs much of this functionality. We recommend that you first become familiar with the CICS PA facilities described in this chapter before you explore Chapter 19, “Historical Database” on page 415.
2.1 What CICS PA is

CICS PA for z/OS helps you tune, manage, and plan your CICS systems effectively. It is a reporting tool that provides about the performance of your CICS systems and applications. Figure 2-1 shows an overview of CICS PA operation.

CICS PA helps you to analyze all aspects of your CICS systems, including:

- CICS application performance
- CICS system resource usage
- Cross-system performance, including multi-region operation (MRO) and advanced program-to-program communication (APPC)
- Transaction groups, including CICS Web Support, Internet Inter-ORB Protocol (IIOP), external call interface (ECI) over Transmission Control Protocol/Internet Protocol (TCP/IP)
- CICS Business Transaction Services (BTS)
- MVS Workload Manager (WLM)
- Exception events that cause performance degradation
- Transaction file and temporary storage usage
- External subsystems, including DB2, IMS (database control (DBCTL)), and WebSphere MQ
- System Logger performance

CICS PA also provides a Historical Database facility to help you manage the performance data for your CICS transactions.

![Figure 2-1 CICS PA overview](image-url)
CICS PA provides an ISPF-based menu-driven screen that helps you to create, maintain, and submit Report Sets for batch processing. A Report Set allows you to define a set of report and extract requests to run as a one-step job with one pass of the input data. You can define any number of Report Sets. You can also include any number of reports and extracts in a single Report Set. Plus, you can select report categories or individual reports for submission independent of the Report Set.

You can use Report Forms to tailor the format and content of reports and extracts. Specify selection criteria to filter the data on the value of particular fields. Object Lists provide a convenient way to specify lists of values under Selection Criteria.

CICS PA produces reports and extracts using data normally collected by your system in MVS SMF data sets. This includes:
- CICS Monitoring Facility (CMF) performance class, exception class, and transaction resource class data written as SMF type 110 records
- DB2 accounting data written as SMF type 101 records
- WebSphere MQ accounting data written as SMF type 116 records
- MVS System Logger data written as SMF type 88 records

The System Definitions facility allows you to define the systems, SMF data files and groups for reporting on. A Take-up facility is provided.

CICS PA can process Record Selection and Cross-System Work extract data sets in a similar way to the SMF data sets.

Export data sets contain extracts of CMF performance data that is suitable for further analysis and graphing by your favorite database or spreadsheet tools.

The HDB Register is an inventory of all information associated with the Historical Database Manager. HDBs save performance data in data sets that are managed from the screen. You can run reports on your HDB or export the HDB data to DB2 tables. Report Forms can be used to control the format and content of the reports. You can specify selection criteria to filter the data that is reported or exported.

There are two types of HDBs:
- **List HDB data set**: One record represents one transaction. Typically, List HDBs are used to analyze recent transaction events. Data is usually only required for a short period of time.
- **Summary HDB data set**: One record represents a summary of transaction activity over a user-specified time interval. Typically, Summary HDBs are used for long-term trend analysis and capacity planning. Data is retained for a longer period of time, sometimes years.

### 2.2 System requirements

Refer to the *CICS Performance Analyzer for z/OS User's Guide*, SC34-6307, for the hardware, software, and storage requirements. In addition, there are several program temporary fixes (PTFs) that are required to support the workloads and CICS PA functions described in this book.
Apply the relevant PTFs in Table 2-1 to support the new CICS Transaction Server monitoring features:

- CMF Transaction resource class data for file and temporary storage queue usage
- Application Naming (DFHAPPL)
- CICS Resource Manager Interface (DFHRMI)

**Table 2-1  PTFs to enable new monitoring features of CICS Transaction Server (TS)**

<table>
<thead>
<tr>
<th>Product</th>
<th>APAR</th>
<th>PTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS TS V2.2</td>
<td>PQ63143</td>
<td>UQ68396, UQ68398, UQ68400</td>
</tr>
<tr>
<td></td>
<td>PQ76701</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PQ76703</td>
<td></td>
</tr>
<tr>
<td>CICS TS V1.3</td>
<td>PQ63141</td>
<td>UQ70905, UQ70908, UQ70913</td>
</tr>
<tr>
<td></td>
<td>PQ76695</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PQ76698</td>
<td></td>
</tr>
</tbody>
</table>

Apply the PTFs in Table 2-2 for CICS Performance Analyzer.

**Table 2-2  PTFs for CICS Performance Analyzer**

<table>
<thead>
<tr>
<th>Product</th>
<th>APAR</th>
<th>PTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS PA V1.3</td>
<td>PQ77980</td>
<td>UQ80393</td>
</tr>
<tr>
<td></td>
<td>PQ79058</td>
<td>UQ81351</td>
</tr>
</tbody>
</table>

Apply the PTFs in Table 2-3 for ISPF.

**Table 2-3  PTFs for ISPF**

<table>
<thead>
<tr>
<th>Product</th>
<th>APAR</th>
<th>PTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPF</td>
<td>OA04921</td>
<td>to be determined</td>
</tr>
</tbody>
</table>

### 2.3 Recommended ISPF setup

The CICS PA screen is an ISPF application that follows Common User Access (CUA) conventions. You can use ISPF standard facilities to customize the screen. This section contains some ISPF setup recommendations to help you use CICS PA efficiently.

**Screen size and scrolling**

Set the screen size in your session parameters to 32 lines. CICS PA screens are optimized for 32 lines, but accommodate 24 lines by scrolling backward (F7) and forward (F8).

**Function keys**

CICS PA uses standard conventions for function keys. You can use the ISPF commands KEYS and KEYLIST to assign alternative functions to the keys. For a list of the CICS PA default settings, select Help -> Keys Help in the action bar or enter KEYSHELP on the command line.
If you are new to CICS PA, ensure that the function keys are displayed at the bottom of the screens. The ISPF command PFSHOW ON|OFF turns on and off the display of the function keys.

**Prompt (F4)**

Input fields with a plus sign (+) to the right of the field, or to the right of the column heading, signify that Prompt is available. To use this facility, position the cursor on the input field and press Prompt (F4). A list of available values is displayed from which you can select one (then press Enter) or more (then press Exit) as appropriate.

**Mouse options**

The CICS PA Report Set panel is a tree structure of report categories and reports. The report categories act as folders that can expand (to show) and collapse (to hide) the reports contained within them. If your terminal emulation permits, configure your mouse options to activate the lightpen function. You can then use the left button of your mouse to click the plus sign (+) to expand and the minus sign (-) to collapse the report categories. Alternatively, you can enter line action S.

**CUA attribute settings**

The CICS PA screen is designed to use the default CUA attributes. However, we recommend that you set the Point-and-Shoot field to easily distinguish Point-and-Shoot fields from other types of fields. You can use the ISPF CUAATTR command to change the attribute settings. As shown in Figure 2-2, we changed Point-and-Shoot to yellow (highlighted in bold). For better distinction, you can also set the highlight attribute to REVERSE (reverse video).

![Figure 2-2 Recommended CUAATTR settings for CICS PA](image)

**Point-and-Shoot fields**

CICS PA employs the Point-and-Shoot field. For efficient use, enter the ISPF SETTINGS command to display the ISPF Settings screen. Then select Tab to point-and-shoot fields (highlighted in bold in Figure 2-3).

**Long and short messages**

CICS PA uses both long and short messages. Short messages display at the top right of the screen on the same line as the screen title. After a short message is displayed, you can press Help (F1) to display more information in a long message.

To display long messages in a pop-up window, enter the SETTINGS command and select Long message in pop-up (highlighted in bold in Figure 2-3). If it is a field in error, the pop-up window displays immediately above or below the field in error. If this option is not selected,
long messages of less than the screen width display immediately above or below the command line.

![ISPF Settings table]

### 2.4 Preparing the SMF data for CICS PA

CICS PA processes non-active SMF data sets. There is no special preparation required for CICS PA other than to dump the active SMF data sets at an appropriate time.

CICS and other subsystems, such as DB2, WebSphere MQ, and the MVS System Logger, write their SMF records to the active SMF data set. In order for CICS PA to work with the inactive copy of this data, you need to perform several steps.

In a CICS region to ensure that all the current SMF data is available, you may need to flush the buffers within CICS that hold any SMF data. You can do this by turning off and on performance monitoring, using the CEMT SET MONITOR command. You only need to flush the buffers when the CICS region is not shut down.

After all the SMF data from the CICS region is on the active SMF data set, you need to dump this data to an inactive SMF data set. First you switch the recording of SMF data from one data set to another. All SMF data in storage is written out before the transfer is made. This switch is performed by issuing the /I SMF operator command. The switch of SMF data sets takes place automatically when the active SMF data set becomes full.

To dump the SMF data set, the SMF dump program (IFASMFDP) is provided. This program transfers the contents of the active SMF data set to an output data set. Then it resets the status of the dumped data set to empty so that SMF can use it again for recording data. CICS PA uses this output data set as the input data for its report processing. See z/OS V1R4.0 MVS System Management Facilities (SMF), SA22-7630, about dumping SMF data.
2.5 Starting CICS PA

The CICS PA screen is invoked when you enter the following command on the ISPF Command Shell screen (option 6) command line:

```sh
ex 'CICSPA13.SCPAEXEC(CPAOREXX)' 'CICSPA13 E'
```

If CICSPA13 is not the high-level qualifier of your CICS PA Release 3 data sets, then alter the command accordingly.

You can also define it as a standard selection on ISPF screens. For examples of how to do this, refer to the “Installation” chapter in the CICS Performance Analyzer for z/OS User’s Guide, SC34-6307.

2.5.1 CICS PA Primary Option Menu

Upon entry to the CICS PA screen, you see the CICS PA Primary Option Menu as shown in Figure 2-4. If you are using CICS PA for the first time, you can select option 0 (CICS PA Profile) to review or modify your default profile settings. This is optional because the CICS PA defaults are sufficient for us to get started. CICS PA allocates new data sets on your behalf when it needs them to save your report requests.

```
VIR3MO            CICS Performance Analyzer - Primary Option Menu
Option ===> __________________________________________________________
0  CICS PA Profile       Customize your CICS PA dialog profile
1  System Definitions    Specify CICS Systems, SMF Files and Groups
2  Report Sets           Request and submit reports and extracts
3  Report Forms          Define Report Forms
4  Object Lists          Define Object Lists
5  Historical Database   Collect and process historical data
X  Exit                  Terminate CICS PA
```

Figure 2-4  CICS PA Primary Option Menu

2.6 System definitions

Before you request CICS PA reports, you must first define the CICS systems (generic APPLIDs) on which you want to report. You also may need to define DB2 subsystems for the DB2 report, MQ subsystems for the WebSphere MQ report, and MVS System Loggers for the System Logger report.

You must also specify the SMF data sets for the systems (CICS, DB2, MQ, Logger), for the MVS System (Image) where they execute, or for both. In addition, you can define groups of systems for reporting purposes, such as those systems that connect via interregion communication/multiregion operation (IRC/MRO) or intersystem communication/advanced program-to-program communication (ISC/APPC).

To specify system definitions, select option 1 from the Primary Option Menu (Figure 2-4).

**Tip:** You can link directly to System Definitions from anywhere in the screen by entering SYSDEFS on the command line.
2.6.1 System Definitions Menu

The first time that you invoke System Definitions, you see the System Definitions Menu (Figure 2-5). From this menu, you can:

- Define systems, SMF files, and groups on which you want to report.
- Maintain SMF files for each system, for each MVS system (Image), or for both.
- Maintain group definitions for reporting purposes.
- Use the data Take-up facility to extract details of your systems from an SMF file for automatic take-up into your system definitions.

You can choose to bypass this menu in the future by selecting “Always go directly to Systems View” as shown in bold at the bottom of Figure 2-5. In this scenario, we select option 4.

```
Figure 2-5  System Definitions Menu

Command ===> ____________________________________________
Select an option then press Enter.

4  1. Define Systems, SMF Files and Groups
   2. Maintain SMF Files
   3. Maintain Group definitions
   4. Take-up from SMF File

Enter "/" to select option
   _ Always go directly to Systems View
```

2.6.2 Take-Up from SMF

An easy way for us to start is to let CICS PA set up our system definitions by using the Take-up facility. This facility populates the system and file definitions with details extracted from SMF files.

Since we selected option 4 on the System Definitions Menu screen, we now see the Data Take-Up from SMF screen as shown in Figure 2-6. Specify the details of the SMF file on which you want to report and then press Enter. CICS PA generates a batch job to extract the take-up details from the SMF data set.

```
Figure 2-6  System definitions: Data Take-Up from SMF screen

Command ===> _____________________________________________________
Specify the SMF File for data take-up and press Enter
Data Set Name . . . 'CICSRS7.SMF110.TESTCASE'_____________________
Specify details if data set is not cataloged:
UNIT . . . . . ________  +       VOLSER . . . ______  +
SEQ Number . . ___ (1 to 255)
Execution Mode:
   2  1. Submit Batch JCL
      2. Edit Batch JCL
```
After the job is submitted, press F3 until you return to the Primary Option Menu (Figure 2-4 on page 29). Again select option 1 (System Definitions). You are now prompted by CICS PA to update your system definitions with the results of the batch job. Figure 2-7 shows the message “CICS PA has completed extracting systems from the following SMF File”, which you receive when the SMF extract is complete.

![Figure 2-7 Populating your system definitions with take-up details](image)

Press Enter to tell CICS PA to populate your system definitions with the details extracted from the SMF file. When complete, the System Definitions Menu is displayed with the message “Take-up was successful”.

**Note:** When you run an initial take-up, CICS PA defines the systems, the files, and the system-file relationships. When you run a second or subsequent take-up for systems that are already defined to CICS PA, then only the SMF files are added. Then you need to define the system-file relationships for the added files yourself if required.

### 2.6.3 Maintain system definitions

We now look at the results of the take-up. From the System Definitions Menu (Figure 2-5), select option 1 to display the System Definitions maintenance screen. This is where you define to CICS PA your CICS Systems (APPLIDs), MVS Images, DB2 and MQ Subsystems, and MVS System Loggers so that:

- They can be requested for report and data extract processing
- The SMF files containing the data can be defined
CICS PA has automatically defined the MVS image, SC66. MVS Image entries are identifiable by "Image" in the Type column and the Image column is blank. APPLIDs are listed with a type of "CICS" and Image SC66.

You can see that the SMF Files System is the image SC66 for all systems. This means that files defined to SC66 are available to all systems defined to that Image. Therefore, you only need to define the files once.

Specifying SMF files is optional. If you do not specify them here, then when it comes time to submit your report request, CICS PA generates job control language (JCL) with the SMF File data set names unresolved. You have the option to edit the JCL at that time.

You can define new systems by entering the NEW command on the command line. Consider the following examples:

```
NEW CICSPAOR CICS
NEW SC43 IMAGE
```

In this example, type line action S next to the SC66 Image entry (highlighted in bold in Figure 2-8) to select it from the System Definitions screen.

### 2.6.4 MVS image definition

Let's look further at the results of the take-up. Now you see the MVS Image display as shown in Figure 2-9. Notice that the item in bold indicates the files for this system.
SMF Files for this system

Observe that the SMF data set name listed is the one specified in the take-up job. You can specify as many files as you want. CICS PA processes them all (unless they are excluded). We recommend that you specify the files in time sequence (earliest first), since CICS PA processes them in the order that they are specified. Various line actions are available to help you do this: I (Insert), R (Repeat), C (Copy), M (Move), and D (Delete).

Deleting a file here only deletes the relationship, not the file itself. Also, you can use the X line action to exclude an SMF File from report processing. Excluded files are marked with an asterisk (*) in the Exc column.

To add a file to the list, you can type the data set name directly, or select from a list of available files by entering line action S or pressing F4 (Prompt) from the data set name field.

Groups this system belongs to

Press F11 to scroll right. More: > is displayed in the top right corner to remind you that there is more information for this system. A screen like the example in Figure 2-10 is displayed where you can specify the groups to which this system belongs on the line under Group + and Description.

Groups enable you to connect systems together for consolidated (cross-system) reporting. This is especially useful for MRO, APPC or other systems that share workloads. For examples of grouping systems, see Figure 7-21 on page 189 and Figure 13-7 on page 289.

Press F3 to return to the System Definitions screen.
2.6.5 CICS System definition

From the list of System Definitions, you can select other system entries to review or modify. For example, select the first CICS system. Then you see a screen like the example in Figure 2-11.

![Figure 2-11 CICS System definition](image)

To define a CICS System for reporting, you only need to specify the APPLID. All other fields are optional.

Notice the MVS Image (SMF ID) to which this CICS System (APPLID) belongs. The MVS Image allows CICS PA to:
- Distinguish between multiple CICS systems that have the same APPLID but run on different MVS Images.
- Share SMF files that contain data for more than one system. By defining the SMF files to the MVS Image, you need only define your SMF files once.
- Request reporting by MVS Image. All CICS Systems (APPLIDs) belonging to that MVS Image are selected.

Tips:
- You can specify a masked pattern for the name of your system. For example, you can define APPLID SCSCP\*. This allows all CICS systems matching this pattern (SCSCPAA1, SCSCPTA1, and so on) to share the System definition, SMF files, and groups specified once for SCSCP\*.
- CICS systems that are not defined to CICS PA can still be reported, but only if their Image is defined. For example, if CICSPFOR (your production file owning region) also runs on Image SC66, then at run report time, you can request reporting for this system. You specify a System Selection of CICSPFOR and SC66, even though CICSPFOR is not defined to CICS PA.

These other fields may be important to you in the future:
- **MCT**: You must specify the monitoring control table (MCT) suffix and MCT load library if you want to include CMF user fields in your reporting. Otherwise, CICS PA uses the system default MCT for the version of CICS you are reporting.
- **Dictionary DSN**: You can build a data set to contain the CMF dictionary record for those times when the SMF file does not contain one, so that CICS PA reporting can progress.
CMF uses a dictionary record to map the fields in the CMF performance class records. CICS writes a dictionary record when CMF starts, but not when SMF switches data sets. You only need to build a dictionary record if you want to include your CMF user fields (from user defined EMPs in the MCT) in your reporting. Otherwise, CICS PA uses the default dictionary record for the version of CICS that you are reporting.

If you want CICS PA to generate the Dictionary record for this CICS system, follow these steps:

a. Specify the Dictionary DSN.

b. Specify the SDFHLOAD Library so that CICS PA can use the DFHMNDUP utility to generate the Dictionary record.

c. Select Dictionary in the action bar. CICS PA immediately populates the specified data set with the Dictionary record for this CICS system. If the data set is not cataloged, CICS PA allocates it before writing the record. If the data set is cataloged, CICS PA overwrites its contents with the new Dictionary record.

At JCL generation time, CICS PA inserts the cataloged Dictionary DSN in the CPADICTR DD statement.

Tip: If you are using an MCT to exclude CMF fields, you do not need to specify the MCT to CICS PA for that reason alone. Check that you did not exclude fields that CICS PA requires for your reporting. Refer to the CICS Performance Analyzer for z/OS Report Reference, SC34-6308, for the list of required CMF fields for:

- Cross-System Work report and extract
- Transaction Group report
- BTS report
- Workload Activity report
- DB2 report

2.6.6 Other system definitions: DB2, WebSphere MQ, System Logger

Other systems are identified by their type, such as DB2, MQ, or LOGGER. To define them to CICS PA, the System Definitions facility is used in a similar way to CICS APPLIDs and MVS Images.

The initial system definition is complete. You can now move on to requesting reports. Press F3 until you return to the Primary Option Menu (Figure 2-4 on page 29).

2.7 Requesting reports and extracts

To build report and extract requests, select option 2 on the Primary Option Menu (Figure 2-4 on page 29).

2.7.1 Creating the Report Sets data set

You are prompted to create the Report Sets data set as shown in Figure 2-12. This is the data set in which CICS PA saves your report and extract requests.
2.7.2 Report Sets

The Report Sets facility defines, maintains, and runs report and extract requests. A Report Set contains a set of report and extract requests to be submitted and run as a single job. You can define as many Report Sets as you want. You can also define any number of reports and extracts in a Report Set. Figure 2-13 shows the list of Report Sets, which initially is empty.

2.7.3 Editing the Report Set

You can now start editing your Report Set shown in Figure 2-14. The list of available reports and extracts is presented as a tree structure where they are grouped by category. You can use line action S to expand and collapse the categories to show or hide the items within it. This is similar to the way some PC tools display folders and their contents. If your terminal emulation allows, you can set your mouse as a lightpen and then click the + to expand or - to collapse the category. Alternatively, you can use cursor selection. Position the cursor on the + or - sign and press Enter.
You can use line action $S$ to select the reports and extracts that you want to edit, and the global options and selection criteria that you want to apply to them. You can also issue line actions on report categories and on "**Reports**" at the top of the tree. Enter the / line action next to an item in the tree to see the list of possible actions.

The selection criteria enables you to filter the CMF data for your reports and extracts using any field or combination of fields. For example, to include data only for a particular transaction ID, user ID, or only for a specific period of time.

Select the global options and then select the Performance List report. Type $S$ next to both options and then press Enter.
2.7.4 Global Options

Figure 2-15 shows the Report Set Global Options. They define general control information that applies to all the reports and extracts in the Report Set.

| Command ===>
<table>
<thead>
<tr>
<th>Preset Options:</th>
<th>REDBOOK - Global Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Selection:</td>
<td></td>
</tr>
<tr>
<td>Report Formatting Options:</td>
<td></td>
</tr>
<tr>
<td>Print Lines per Page . . 60_ (1-255)</td>
<td></td>
</tr>
<tr>
<td>Time Zone . . . . . . . ___ (Blank for system default or -12 to +12 hours)</td>
<td></td>
</tr>
<tr>
<td>Date Delimiter . . . . /</td>
<td></td>
</tr>
<tr>
<td>Time Delimiter . . . . :</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2-15  Reviewing Global Options

Note the following points:

- To specify the systems (and inherently the files) on which you want to report, you can specify:
  - A CICS APPLID
  - A DB2 subsystem ID
  - A WebSphere MQ subsystem ID
  - An MVS System Logger ID
  - An MVS Image ID
  - A Group ID

  You can type the IDs directly or use Prompt (F4) to select from a list of predefined systems and groups.

- You can specify System Selection in one or more of the following ways:
  - As a global option on the Global Options screen.
  - As a local option on individual report or extract screens. Report-level specifications take precedence over the global specification.
  - As a global option on the Run Report Set screen. The run-time global option overrides the Report Set global option and optionally the report-level specifications.
  - By editing the JCL before submit.

- Print Lines per Page is the maximum number of report lines to print on each page. The default is 60. You can also specify this option for individual reports. Report-level specifications take precedence over the global specification.

- Date and Time delimiters specify the separator characters for the date and time-of-day in the reports and extracts. A slash (/) and a colon (:) are the defaults.

- The Time Zone specifies the number of hours east or west of Greenwich Mean Time (GMT). For example, to synchronize the CMF and DB2 time stamps, specify Time Zone to match the time zone of the SMF data. However, if you are correlating DB2 data between CICS PA and DB2 PM reports, then you may want the CICS PA DB2 time stamps to be reported in GMT so that they can be more easily matched. If Time Zone is not specified, or it is set to zero, all times (CMF and DB2) are reported in GMT.
In this case, accept the default global options. Now you can exit or cancel to continue.

### 2.7.5 Specifying report options

The Performance List report gives the details of every transaction that executed. Figure 2-16 shows the screen where the Performance List report options are specified.

The report runs without you specifying any additional options. However, you may want to tailor it to help your analysis. If you want to specify a particular system that this report applies to, under System Selection, enter the System name (and optionally Image, Group, or both). Alternatively, you can select the required system from a list. To do this, position the cursor on the APPLID field (highlighted in bold) and press F4 (Prompt).

![Figure 2-16 Specifying report options: Using F4 (Prompt) to select from a list of systems](image)

A selection list of available systems is displayed as shown in Figure 2-17. Enter line action S (as shown in bold) to select the system that you want.

![Figure 2-17 Selecting a system](image)

Then CICS PA sets the information under System Selection as shown in Figure 2-18.
If you decide not to specify the System Selection here, then you can do so when you run your Report Set and CICS PA prompts you.

Two important report options that we discuss later are the report format and selection criteria. They allow you to tailor the fields that appear in your reports and filter the data that is reported.

Exit to save your new report request.

### 2.7.6 Reports list

After you exit from the report, the Reports list is presented (Figure 2-19). You can define as many reports of the same type in a Report Set as you want.

You can use line action I (Insert) to define a new Performance List report, D to delete a report, or X to exclude it from reporting. When you finish defining your Performance List reports, exit to save the reports and return to the main Report Set edit screen.

![Figure 2-19 Reports list](image)

### 2.8 Running your reports

You must enter a SAVE command, select File->Save from the action bar, or press F3 (Exit) to save your Report Set definition in the Report Set data and exit. However, you do not need to save your Report Set before you run it.

To run the Report Set, enter the RUN command in the command line or select File->Run from the action bar. Alternatively, you can run individual reports or report categories by
enters the RUN line action next to the particular ones that you want to run. Figure 2-20 shows both methods.

2.8.1 Active status

Looking at the Report Set in Figure 2-20, you can see that the Performance List report and the Performance Reports category are active. Observe also that the Global Options are automatically activated by CICS PA when there is at least one active report or extract.

The Active status controls which reports in the Report Set are run when you submit a RUN request. When you enter RUN in the command line to run the Report Set, only active reports within active categories are selected for JCL generation. You can temporarily override the active status by typing the RUN line action next to any required reports and categories.

You can use line action D to deactivate or A to activate particular reports and categories.

2.8.2 Run Report Set

Before CICS PA generates the JCL, you are prompted to supply run-time options as shown in Figure 2-21.
You can specify the following run-time options:

- **The system or group of systems to be reported:** CICS PA allows you to specify System Selection both in the Report Set and here at run-time. An Override System Selections option is provided to determine which specification takes precedence if both are specified:
  - When the override option is not selected, the run-time System Selection overrides the Report Set Global options only. It does not override any System Selections specified in the individual reports within the Report Set.
  - When the override option is selected, the run-time System Selection overrides all System Selections in the Report Set (Global Options and individual reports).

- **The date and time range or “time slot” of the SMF data that you want to process:** This reduces the volume of data and enables more efficient processing. If not specified, then CICS PA processes the entire SMF file or files. Any report intervals specified under Selection Criteria in your Report Set are then processed normally for this reduced period of data.

- **Missing SMF Files Option:** This specifies the remedial action to take if you have not defined SMF files for the systems to be reported (or they are all excluded).

- **To edit the JCL before submit**

**Note:** Because we specified System Selection on the Performance List Reports screen (Figure 2-19 on page 40), specifying System Selection now at run-time has no effect on that report unless we select the Override System Selections option.

From the Run Report Set screen, press Enter to generate the JCL.

### 2.8.3 JCL generation

Now you see an ISPF Edit session with the JCL as shown in Figure 2-22. You can store the job stream in your JCL library to submit from there, or as part of a job automation process.
Make any necessary changes. Then type SUBMIT (or SUB) on the command line and press Enter to submit the job.

Figure 2-22  Edit the Report Set JCL and submit the job

After you submit the job, exit until you return to the CICS PA Primary Option Menu (Figure 2-4 on page 29).

2.9 Viewing the report output

You can view the output using SDSF or ISPF option 3.8 Outlist Utility (from the Primary Options Menu, option 3 and then option 8). The CICS PA screen automatically assigns each report in the Report Set a unique DDname. In SDSF, you can each report separately by entering the question mark (?) action character in the NP column (Figure 2-23).

Figure 2-23  SDSF: Displaying the job output by DDname

Then enter the S action character to select your report output (Figure 2-24).
2.10 Tailoring report formats

Report Forms allow you to design your own reports and extracts to fully exploit the wealth of information contained in the CICS Transaction Server (TS) CMF performance records. For example, if you suspect that there is a performance problem with transient data, you can create a Report Form that focuses on that aspect of CICS performance.

To build Report Forms, select option 3 from the Primary Option Menu (Figure 2-4 on page 29).

2.10.1 Creating the Report Forms data set

You are prompted to create the Report Forms data set. CICS PA saves your Report Forms in this data set.

Press Enter to create the Report Forms data set. Otherwise, cancel and from the Primary Option Menu (Figure 2-4 on page 29), select 0.4 (option 0 and then option 4) to specify the data set name of your choice.

2.10.2 Report Form types

There are three types of Report Forms: LIST, LISTX and SUMMARY. You can tailor your reports and extracts using Report Forms of a compatible type:

- **LIST**: Specifies which fields are reported and the order of the columns. This type of form is applicable to:
  - Performance List report
  - Cross-System Work (Extended) report
– Export extract
– List HDB reports

- **LISTX**: Specifies which fields are reported, the order of the columns, up to three sort fields (ascending or descending), and a processing limit on one of the sort fields. This type of form is applicable to:
  – Performance List Extended report
  – Cross-System Work (Extended) report (sort sequence and limit ignored)
  – Export extract (sort sequence and limit ignored)

- **SUMMARY**: Specifies which fields are reported, the order of the columns, up to three sort fields (ascending), and numeric functions (average, standard deviation, total, minimum, maximum). This type of form is applicable to:
  – Performance Summary report
  – Export extract
  – Summary HDB reports

### 2.10.3 Report Forms list

The Report Forms list displays all the Report Forms in the Report Forms data set and shows their type and description. The initial Report Forms list is empty as shown in Figure 2-26. You can use the NEW command to create your own Report Form or you can select from the many samples provided. We use the samples.

To display the list of Sample Report Forms, enter the SAMPLES command or select **Samples** in the action bar as shown in Figure 2-26.

![Figure 2-26 Report Forms list: Requesting Sample Report Forms](image)

### 2.10.4 Sample Report Forms

On the Sample Report Forms screen (Figure 2-27), scroll down until you find the Sample Report Forms that meet your requirements.

We selected two Report Forms for Transient Data Analysis:
- **TDLST** lists all transactions, showing their Transient Data usage
- **TDSUM** summarizes Transient Data usage for each transaction ID
Before we finish, let's look at Report Form TDSUM to familiarize ourselves with the format of the report it will produce and introduce some of the features. Enter line action S to select TDSUM.

---

### Sample Report Forms

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDLST</td>
<td>LIST</td>
<td>Transient Data Activity</td>
</tr>
<tr>
<td>TDSUM</td>
<td>SUMMARY</td>
<td>Transient Data Activity</td>
</tr>
<tr>
<td>TRAPLSUM</td>
<td>SUMMARY</td>
<td>Transactions by Application Tran</td>
</tr>
<tr>
<td>TRARTSUM</td>
<td>SUMMARY</td>
<td>Transaction Routing Analysis (3)</td>
</tr>
<tr>
<td>TRATDSUM</td>
<td>SUMMARY</td>
<td>Transactions by Applid and TOD</td>
</tr>
<tr>
<td>TRTRESUM</td>
<td>SUMMARY</td>
<td>Transaction Routing Analysis (1)</td>
</tr>
<tr>
<td>TRTCLSUM</td>
<td>SUMMARY</td>
<td>Transactions by Tranclass Name</td>
</tr>
<tr>
<td>TRTESUM</td>
<td>SUMMARY</td>
<td>TransactionUsage by Terminal ID</td>
</tr>
<tr>
<td>TRTODSUM</td>
<td>SUMMARY</td>
<td>Transactions by Time-of-Day</td>
</tr>
<tr>
<td>TRTRASUM</td>
<td>SUMMARY</td>
<td>Transaction Routing Analysis (4)</td>
</tr>
<tr>
<td>TSLSUM</td>
<td>LIST</td>
<td>Temporary Storage Activity</td>
</tr>
<tr>
<td>TSSUM</td>
<td>SUMMARY</td>
<td>Temporary Storage Activity</td>
</tr>
<tr>
<td>TSWTLST</td>
<td>LIST</td>
<td>Temporary Storage Wait Analysis</td>
</tr>
<tr>
<td>TSWTSUM</td>
<td>SUMMARY</td>
<td>Temporary Storage Wait Analysis</td>
</tr>
<tr>
<td>UOWLST</td>
<td>LIST</td>
<td>Transaction Network Unit-of-Work</td>
</tr>
<tr>
<td>USTORLST</td>
<td>LIST</td>
<td>User (Task) Storage Analysis</td>
</tr>
</tbody>
</table>

---

**Figure 2-27** Selecting Sample Report Forms

Exit to add these Report Forms to your Report Forms data set. You can see them in the Report Forms list in Figure 2-28. They are now available for report processing.

2.10.5 **Edit Report Form**

You can now review or change the Report Form as shown in Figure 2-29.
You can specify selection criteria in the Report Form. If a report that uses this form also has selection criteria defined, then records are selected for reporting only if they satisfy both criteria.

If the Report Form does not meet your reporting requirements, you can change it so that only the fields you require in your report are above the EOR line (limited to a page width of 132), and only those you require in your extract are above the EOX line (no limit).

The line actions that you can use to edit the Form include I (Insert), D or DD (Delete), C or CC (Copy), M or MM (Move), R or RR (Repeat).

To add another field to the Form, you can either replace an existing field by overtyping it, or you can use line action I to insert a blank row to accept the new field. You can either type the field name (or first part of it) directly, or you can enter line action S to select the field name from a list of allowable fields.

We entered line action I (Insert) to add a new field named RESPONSE into the Form.

You can enter a FIND command to locate a character string in the display. Then press F5 or use the RFIND command to locate the next occurrence.

You can enter line action H (Help) to obtain a detailed description of a field.

More: > indicates that you can scroll Right (F11) to view more information. This includes field length, Dictionary definition, User Field offset and length, and report title. The title appears at the top of each page of the report immediately below the date and time. The first line of the specified title appears on the left of the report, and the second line on the right. You can also specify a Title for individual reports. This takes precedence over that in the Report Form.
Our form indicates that:

- The report is summarized by TRAN, the transaction ID.
- Nine fields are shown in the report, from TRAN in the left-most report column to TDWAIT in the right-most column.
- Statistical averages for RESPONSE, TDGET, TDPURGE, and so on, are reported.
- TDWAIT is reported in two columns. TIME shows the average I/O Wait elapsed time. COUNT shows the average number of times transactions waited for TD.
- EOR indicates where the report line ends. CICS PA automatically adjusts this for you to ensure that the fields you specify fit across the page.
- EOX indicates where the extract record ends. There is no restriction on the record length. You can move EOX to the bottom of the Report Form to include all available fields in the extract.

Exit (F3) from the Report Form to save it.

### 2.10.6 New Report Form

From the Report Forms list, you can create a new Report Form by entering the NEW command or selecting **File->New** from the action bar. The New Report Form screen is displayed as shown in Figure 2-30.

![Figure 2-30 New Report Form](image)

**Note:** If you want to include user fields in your Report Form, you must specify the APPLID so that CICS PA can obtain the associated dictionary entries. Otherwise, simply specify the version (VRM) so that CICS PA can populate the form with the fields that are applicable to that release of CICS.

**Report Form field categories**

When creating a Report Form, you can select fields from *all* the CMF data fields or just from specific field categories.

On the New Report Form screen, enter line action `S` next to Field Categories to display the list of categories defined in CICS PA (Figure 2-31).
Enter the SELECT command to select all categories (the default), or line action S to select particular categories. Then press F3 (Exit).

When all options on the New Report Form screen are specified, press Enter to proceed with creating the Form. Edit the Report Form as required (see Figure 2-29 on page 47). Then exit to save the form.

Exit Report Forms and return to the Primary Option Menu (Figure 2-4 on page 29).

### 2.10.7 Using the Report Form in your report

To use the Report Forms in your report requests, again select option 2 (Report Sets) from the Primary Option Menu (Figure 2-4 on page 29). The list of Report Sets is displayed as shown in Figure 2-32. Enter line action S to resume editing your Report Set.

A Report Set can include more than one report of each type. For example, you can request two List reports and three Summary reports.

Select the reports that you want to edit. Using the S line action as shown in Figure 2-33, we select the Performance List and Summary reports so that we can use our Report Forms.
Since we previously defined one List report, CICS PA presents the list of reports for you to review or update (Figure 2-34). You can add, delete, or exclude reports, or select reports to modify their options.

You can modify some options directly on this screen. We specify the Report Form now. You can either type the name, or press F4 (Prompt) to select a name from the list of available Report Forms for this type of report.

This is all we need to do for the Performance List report. Now press F3 (Exit).

CICS PA continues to the next selected report, in this case the Summary report. This is our first Summary report, so we specify the report options (Figure 2-35).
Chapter 2. A quick start to CICS Performance Analyzer

2.11 Filtering the report

You can specify selection criteria to filter the input records so that your CICS PA reports and extracts only include the data that you interested in. Exception selection criteria applies to the Exception reports and specifies the filtering options for CMF exception class records. performance selection criteria applies to all the other reports and extracts, except the System Logger report. It specifies filtering options for CMF performance class and transaction resource class records, and where applicable, DB2 and WebSphere MQ accounting records.

You can specify global selection criteria that applies to all the reports and extracts in a Report Set, or local selection criteria that applies to an individual report or extract. You can also specify selection criteria in Report Forms.
We select the global performance selection criteria as shown in Figure 2-36 so that we can filter both of our reports to see only the transactions that we are interested in.

2.11.1 Selection criteria

Selection criteria enables you to specify report filtering options as shown in Figure 2-37.

Note the following points:

- You can specify one or more report intervals. Transactions that either start, stop, or are active during the report intervals can be included (INC) or excluded (EXC) from the report.
You can specify one or more fields and a single value, a masking pattern (for character fields), a range of values (for numeric fields), or an Object List (see 2.12, “Maintaining Object Lists” on page 56). Records with data fields that match the specified values can be included (INC) or excluded (EXC) from the report.

For character fields, the masking characters % and * are allowed as well as the ability to select null fields by specifying ' ' (two single quotes).

For numeric fields (Count, Time, or Clock), you can precede the From value with a comparison operator. For example, specify >=1 for a comparison of greater than or equal to 1. Allowed operators are:

=  >  >=  <  <=

Specify time values in seconds using a decimal point. Otherwise, milliseconds is assumed. For example, specify 1.12 seconds or 1120 milliseconds.

You can scroll right by pressing F11 to see more columns of information about the fields such as length and dictionary definition.

You can specify most of the CMF fields in selection criteria. Enter line action S to select from a list of available fields or press F4 (Prompt) on the Field Name. Figure 2-38 shows the field selection list.

You can enter a FIND command to locate a character string in the display. Then use F5 or the RFIND command to locate the next occurrence.

You can enter line action H (Help) to obtain a detailed description of a field.

You can scroll right by pressing F11 to view more columns of information about the fields (Dictionary definition).

We will the field TDWAIT, and then type line action S (Select) and press Enter to insert this field into the select statement.

Complete the select statement to ensure that we only report our Finance transactions (transaction IDs that start with FIN) that waited for at least one Transient Data request.

Press F3 (Exit) to save your select statement. Figure 2-39 shows the select statements (rows) that define your selection criteria.
Selection criteria is defined by one or more select statements. You can add (A), delete (D), or exclude (X) select statements, or select (S) any to modify the specification.

Our specification is complete, so press F3 (exit) to save the selection criteria. Observe in Figure 2-40 that the global performance selection criteria is now active. Enter the RUN command to run the Report Set.

2.11.2 Run Report Set

When the Run Report Set screen is displayed, review the run-time options and press Enter. This time the message System not specified is displayed. Press Help (F1) to display the long error message:

CPA1028E Report Set JCL generation failed. System or Group not specified
CPA1030E System=N/A, Report=Performance Summary, Output=SUMM0001.

This indicates that CICS PA needs to know on which system to run the Summary report.

CICS PA positions the cursor at the System Selection CICS APPLID field. You simply need to press F4 (Prompt) to display the list of available Systems. Select the desired System (SCSCPA1/SC66) and press Enter to insert into your System Selection.

The Run Report Set specification is complete, so press Enter to proceed with JCL generation.
2.11.3 JCL generation

The Report Set JCL is similar to the JCL in Figure 2-22 on page 43. The difference is that additional commands are generated to honor the selection criteria and report forms that you since specified. This is shown in Figure 2-41.

Make any necessary changes. Then type SUBMIT (or SUB) on the command line and press Enter to submit the job.

```
*** Command Input
//SYSIN DD *
* Report Set =REDBOOK
   CICSPA IN(SMFIN001),
      SELECT(PERFORMANCE(INC(TRAN(FIN*)), EXC(TDWAIT(COUNT(0)))), LIST(OUTPUT(LIST0001),
                     FIELDS(TRAN, USERID, TASKNO, STOP(TIMET), TDGET, TDPURGE, TDPUT, TDTOTAL, TDWAIT(TIME), TDWAIT(COUNT)),
                     TITLE1('Transaction Transient Data Activity - Detail ')),
      SUMMARY(OUTPUT(SUMM0001),
                     BY(TRAN),
                     FIELDS(TRAN, TASKCNT, RESPONSE(AVE), TDGET(AVE), TDPURGE(AVE), TDPUT(AVE), TDTOTAL(AVE), TDWAIT(TIME(AVE)), TDWAIT(COUNT(AVE))),
                     TITLE1('Transaction Transient Data Activity - Summary ')),
      /*
```

Global selection criteria
Performance List report
Report Form TDLST
Performance Summary report
Report Form TDSUM

* Figure 2-41  Report Set JCL: Showing the selection criteria and report forms

After you submit the job, press Exit until you return to the CICS PA Primary Option Menu (Figure 2-4 on page 29).
2.12 Maintaining Object Lists

Let us now extend our use of selection criteria by employing an Object List. Object Lists enable you to define a group of related values once. Then you simply refer to the Object List name when specifying the record selection criteria in your Report Sets.

To define Object Lists for use in selection criteria, select option 4 from the Primary Option Menu (Figure 2-4 on page 29).

2.12.1 Creating the Object Lists data set

You are prompted to create the Object Lists data set. This is the data set in which CICS PA saves your report and extract requests.

Press Enter to create the Object Lists data set. Otherwise, cancel and from the Primary Option Menu (Figure 2-4 on page 29), select option 0.5 (option 0 and then option 5) to specify the data set name of your choice.

2.12.2 Object Lists

The Object Lists facility is used to create, modify, and view Object Lists. An Object List defines a list of field values that can be used when specifying record selection criteria, for example, to define all the transaction IDs that belong to a particular application system.

Object Lists enable you to define a group of related values once. Then you simply refer to the Object List name when specifying the record selection criteria in a Report Set. You can define your Object Lists hierarchically to eliminate duplication and improve the integrity of lists.

The initial list of Object Lists is empty. Use the NEW command to create your first Object List. An Object List is a member in the Object Lists data set.

2.12.3 Edit Object List

You can now start editing your Object List. The example in Figure 2-42 shows a list of long running CICS internal transactions that you may commonly want to exclude from your reporting. For other examples of Object Lists, see Figure 13-44 on page 309 and Figure 18-6 on page 391.

In any row, you can specify:
- A single value
- A pattern using masking characters % and * (character fields only)
- A range (numeric fields only)
- An Object List (sublist)

You can specify any number of values in an Object List. You can also specify any number of Object Lists as sublists.

The order of entries in the list is of no consequence to CICS PA reporting.
2.12.4 Using the Object List in your selection criteria

To use the Object List in your selection criteria, follow these steps:

1. From the Primary Option Menu, select option 2 (Report Sets).
2. From the Report Sets list, select the Report Set that you want to edit.
3. From the Edit Report Set screen, select the Global selection criteria. Or, instead you may specify selection criteria for individual reports.
4. Specify the name of your Object List, which is highlighted in Figure 2-43. You can type the name directly, or press F4 (Prompt) to select from a list of available Object Lists.
5. Exit to save your selection criteria.
6. Run the Report Set or report as appropriate.
This completes the introduction to CICS PA. To learn about the many additional features of CICS PA, refer to:

- Part 2, “CICS Performance Analyzer in action” on page 127, where the screen is used in particular scenarios
- The CICS PA online Help and Tutorial
Reports and extracts

CICS Performance Analyzer (PA) provides a comprehensive suite of reports and extracts to help you analyze and tune the performance of your CICS systems. This chapter describes the purpose of each report and extract, the options that you can specify to tailor the output, and examples for you to see. For detailed descriptions of all the reports and extracts, refer to the CICS Performance Analyzer for z/OS Report Reference, SC34-6308.

To see how to use different tools to analyze the extract data produced by the CICS PA Export facility, refer to Chapter 4, “Processing extracts” on page 107.

Part 2 of this book draws it all together. It discusses a variety of scenarios that employ many of the CICS PA reports and extracts. It also describes how to use the CICS PA Historical Database (HDB) facility to maintain your CICS performance data.
3.1 CICS PA Report Set

A Report Set contains your report and extract requests. The CICS PA screen presents the Report Set in a tree structure, as shown in Figure 3-1, where the available reports and extracts are grouped by category.

```
EDIT                      Report Set - REDBOOK                     Row 1 of 34
Command ===>                                                  Scroll ===> CSR

Description . . . CICS PA Report Set

Enter "/" to select action.

---
** Reports **                           Active
  - ___ Options                                    No
    - ___ Global                                    No
  - ___ Selection Criteria                         No
    - ___ Performance                                No
    - ___ Exception                                  No
  - ___ Performance Reports                        No
    - ___ List                                      No
    - ___ List Extended                              No
    - ___ Summary                                    No
    - ___ Totals                                     No
    - ___ Wait Analysis                              No
    - ___ Cross-System Work                          No
    - ___ Transaction Group                         No
    - ___ BTS                                        No
    - ___ Workload Activity                          No
  - ___ Exception Reports                           No
    - ___ List                                      No
    - ___ Summary                                    No
  - ___ Transaction Resource Usage Reports          No
    - ___ File Usage Summary                        No
    - ___ Temporary Storage Usage Summary           No
    - ___ Transaction Resource Usage List           No
  - ___ Subsystem Reports                           No
    - ___ DB2                                        No
    - ___ WebSphere MQ                               No
  - ___ System Reports                              No
    - ___ System Logger                              No
  - ___ Performance Graphs                          No
    - ___ Transaction Rate                           No
    - ___ Transaction Response Time                  No
  - ___ Extracts                                    No
    - ___ Cross-System Work                          No
    - ___ Export                                     No
    - ___ Record Selection                           No
** End of Reports **
---
```

Figure 3-1 CICS PA Report Set showing available reports and extracts

This chapter discusses the reports and extracts in the same sequence as they are presented in the screen.
Performance reports
The Performance reports are produced from CICS Monitoring Facility (CMF) performance
class data. The reports in this category are:

- Performance List report
  This is a detailed listing of the CMF performance class data.
- Performance List Extended report
  This report provides a sorted, detailed listing of the CMF performance class data.
- Performance Summary report
  This report summarizes the CMF performance class data.
- Performance Totals report
  This report provides totals and averages of the CMF performance class data.
- Wait Analysis report
  This report breaks down wait activity by transaction ID (or other ordering fields). You can
  see at a glance which CICS resources are causing your transactions to be suspended.
  This report can help you to quickly identify the possible source of a performance response
time problem.
- Cross-System Work report
  This report is a detailed listing of segments of work performed by a single CICS system or
  multiple CICS systems via transaction routing, function shipping, or distributed transaction
  processing on behalf of a single network unit-of-work ID. It provides a consolidated report
  that shows the complete transaction activity across connected systems. The format can
  be tailored to produce the Cross-System Work Extended report.
- Transaction Group report
  This report offers a detailed listing of segments of work performed by the same or different
  CICS systems on behalf of a single transaction group ID.
- BTS report
  This is a detailed listing that shows the correlation of the transactions performed by the
  same or different CICS systems on behalf of a single CICS Business Transaction Services
  (BTS) process.
- Workload Activity report
  This report provides a transactions response time analysis by MVS Workload Manager
  (WLM) service and report class. You can use this in conjunction with the z/OS Resource
  Measurement Facility™ (RMF) workload activity reports to understand from a CICS
  perspective how well your CICS transactions are meeting their response time goals. The
  Workload Activity List report is a cross-system report that correlates CMF performance
  class data from single or multiple CICS systems for each network unit of work. Importantly,
  this report ties multiregion operation (MRO) and function shipping tasks to their originating
  task so that their impact on response time can be assessed. The Workload Activity
  Summary report summarizes response time by WLM service and report classes.

Exception reports
Exception reports are produced from CMF exception class data. The reports in this category
are:

- **Exception List**: A detailed listing of the CMF exception class data
- **Exception Summary**: A summary of the CMF exception class data
**Transaction Resource Usage reports**
The Transaction Resource Usage reports are produced from CMF performance class and transaction resource class data. Currently, file and temporary storage usage are the only types of transaction resource data available. The reports in this category are:

- **File Usage Summary report**
  This report provides a detailed analysis of CMF transaction resource class data for files. The Transaction File Usage Summary report summarizes file usage by transaction ID. For each transaction ID, it gives transaction identification and file control statistics followed by a breakdown of file usage for each file used by the transaction. The File Usage Summary report summarizes file activity. For each file, it gives a breakdown of file usage by transaction ID.

- **Temporary Storage Usage Summary report**
  This report provides a detailed analysis of CMF transaction resource class data for temporary storage queues. The Transaction Temporary Storage Usage Summary report summarizes temporary storage usage by transaction ID. For each transaction ID, it gives transaction identification and temporary storage control statistics followed by a breakdown of temporary storage usage for each temporary storage queue used by the transaction. The Temporary Storage Usage Summary report summarizes temporary storage activity. For each temporary storage queue, it breaks down temporary storage usage by transaction ID.

- **Transaction Resource Usage List report**
  This report provides a detailed list of CMF transaction resource class data. The records are reported in the sequence that they appear in the system management facility (SMF) file. The report gives transaction information together with statistics of file storage usage, temporary storage usage, or both by the transaction.

**Subsystem reports**
The Subsystem reports are produced from subsystem accounting data stored in SMF files. The reports in this category are:

- **DB2 report**
  This report correlates CICS CMF records and DB2 Accounting (SMF 101) records by network unit of work to produce a consolidated and detailed view of DB2 usage by your CICS systems. The DB2 report enables you to view CICS and DB2 resource usage statistics together in a single report. The DB2 List report shows detailed information of DB2 activity for each transaction. The DB2 Summary reports summarize DB2 activity by transaction.

- **The WebSphere MQ report**
  Processes WebSphere MQ SMF accounting (SMF 116) records to produce a detailed view of WebSphere MQ usage by your CICS systems. The WebSphere MQ List reports display, depending on the WebSphere MQ accounting traces that are active, details about Transactions, WebSphere MQ Queues that were referenced, WebSphere MQ global (not transaction-specific or queue-specific) statistics and WebSphere queue-specific commands issued by transactions. These can be aggregated by transaction ID, queue name, or both.

**System reports**
The System reports are produced from system data stored in SMF files. The report in this category is the System Logger report.
The System Logger report processes MVS System Logger (SMF 88) records. It does not process CMF data and the selection criteria does not apply. The report provides information about the System Logger log streams and coupling facility structures that are used by CICS Transaction Server (TS) for logging, recovery, and backout operations. The report can assist with measuring the effects of tuning changes and identifying logstream or structure performance problems. The System Logger List report shows information about logstream writes, deletes, and events, as well as Structure Alter events for each SMF recording interval.

The System Logger Summary report summarizes logstream and structure statistics so you can measure the System Logger performance over a longer period of time. These reports, when used in conjunction with the CICS Logger reports produced from the standard CICS statistics reporting utilities, provide a comprehensive analysis of the logstream activity for all your CICS systems.

**Performance Graph reports**

Performance Graph reports are produced from CMF performance class data in graphical format. The reports in this category are:

- **Transaction Rate Graph report**
  
  This report is a set of two graphs that illustrate the average response time and the number of transactions that completed in a specified time interval.

- **Transaction Response Time Graph report**
  
  This report shows a set of two graphs that illustrate the average and maximum response time for all transactions that completed in a specified time interval.

**Extracts**

Performance extracts produce extract data sets from CMF performance class data and create extract data sets in a format that is appropriate to their function. A Recap report is always produced to summarize the extract results. The extracts in this category are:

- **Cross-System Work extract**
  
  This data set is useful for cross-system analysis. CICS PA allows you to merge CMF performance class data from segments of work performed by the same or different CICS systems via transaction routing, function shipping, or distributed transaction processing on behalf of a single network unit-of-work ID. You can use this Cross-System Work data set as input to CICS PA Performance reports such as the List, Summary, and Totals reports to monitor the total amount of resources used by a transaction within a single or across multiple CICS systems.

- **Export extract**
  
  This data set is a subset of the CMF performance class data, extracted and formatted as a delimited text file. This data file can then be imported into PC spreadsheet or database tools such as Lotus 1-2-3 or Lotus Approach for further reporting and analysis. The extract records have a default format that includes all the clock fields. Or you can tailor the format like the Performance List or Performance Summary reports.

- **Record Selection extract**
  
  This data set contains a small extract file with only the records that are of interest to you. The Record Selection Extract filters large SMF files, which can then be used as input to CICS PA. The reduced data volume enables more efficient reporting and analysis. The following record types are processed by this extract:
  - SMF 110 CMF performance records
  - SMF 101 DB2 accounting records, if requested
  - SMF 116 WebSphere MQ accounting records, if requested
3.2 CICS PA commands

CICS PA provides both an Interactive System Productivity Facility (ISPF) screen and a command interface. The CICS PA screen generates the batch job and commands to produce reports and extracts. CICS PA allows you to edit the generated JCL and commands before you submit a job.

The general format of the command as it appears in the SYSIN DD statement of your job is:

\[ \text{CICSPA operand(suboperand),...} \]

3.2.1 Commands for reports and extracts

The operands to request the CICS PA reports and extracts are:

- **LIST** Performance List report
- **LISTX** Performance List Extended report
- **SUMMARY** Performance Summary report
- **TOTAL** Performance Totals report
- **WAITANAL** Performance Wait Analysis report
- **CROSS** Cross-System Work report and extract
- **TRANGROUP** Transaction Group report
- **BTS** CICS Business Transaction Services report
- **WORKLOAD** Workload Activity report
- **LISTEXC** Exception List report
- **SUMEXC** Exception Summary report
- **RESUSAGE(FILESUM)** File Usage Summary report
- **RESUSAGE(TSSUM)** Temporary Storage Usage Summary report
- **RESUSAGE(TRANLIST)** Transaction Resource Usage List report
- **DB2** DB2 report
- **MQ** WebSphere MQ report
- **LOGGER** System Logger report
- **GRAPH(TRANRATE)** Transaction Rate Graph report
- **GRAPH(RESPONSE)** Transaction Response Time Graph report
- **EXPORT** Exported Performance Data extract
- **RECSLE** Record Selection extract

You can specify global operands to apply to all reports and extracts. You can also specify report-specific operands to tailor individual reports and extracts to your particular needs.

3.2.2 Commands for HDB processing

The HDB facility is driven from the CICS PA screen. It has three associated batch processes. The operands are:

- **HDB(LOAD(hdbname))**: Load HDB container data sets with selected SMF performance data
3.3 Performance reports

The reports in this category are:

- Performance List report
- Performance List Extended report
- Performance Summary report
- Performance Totals report
- Wait Analysis report
- Cross-System Work report
- Transaction Group report
- BTS report
- Workload Activity report

3.3.1 Performance List report

The Performance List report provides a detailed list of the CMF performance class records. Figure 3-2 shows the options for this report.

![Figure 3-2 Performance List report options](image)

**Default report format**

Observe the columns of data in the Performance List report in Figure 3-3. This example shows the default format of the report when a Report Form is not specified. It details performance-related information for each transaction.
You can easily change the format of the Performance List report by using a Report Form to display the performance related data in which you are interested. Many sample Report Forms of type LIST are provided with CICS PA for this purpose. The EOR marker in the Report Form defines the end of the print line. It must not exceed the maximum page width of 132. For more information about how to use Report Forms, see 2.10, “Tailoring report formats” on page 44.

Figure 3-4 shows how you can tailor the Performance List report using a Report Form. This example shows File Request activity for each transaction. Notice the File Request counts on the right side of the report.

Example 3-1 shows how you can use the FIELDS operand to tailor the Performance List report.

Example 3-1 Using commands to tailor the Performance List report

```
CICSPA LIST(FIELDS(TRAN, DBCTL(PSBNAME),
    RESPONSE, CPU, IMSREQCT, IMSWAIT(TIME,COUNT),
    DBCTL(SCHTELAP,
    POOLWAIT,
    INTCWAIT,
    DBIOELAP,
    PILOCKEL,
    THREDCPU,
```
Figure 3-5 shows the resulting report with transaction database control (DBCTL) usage.

The DBCTL User Field is 256 bytes long and contains a wealth of IMS information that can be requested in your reports. This information includes:

- PSB name
- Various IMS DBCTL internal elapsed times
- Various IMS DBCTL CPU times
- DLI and database call counts, including DEDB statistics
- Enqueue statistics

### List Export

You can also write your Performance List report to an extract data set. You do this by using the Export facility with a LIST or LISTX Report Form to define the record layout. When you use LISTX, the sort is ignored. The EOX marker in the Report Form defines the end of the extract record. There is no limit on the record length, so you can export all available fields or a selection. For more information, see 3.9.2, “Export extract” on page 101.

#### 3.3.2 Performance List Extended report

The Performance List Extended report is similar to the Performance List report but allows you to sort the data. For example, you can specify:

- The transactions that have the longest response time
- The transactions that use the most CPU time
- The transactions that performed the most File requests

Figure 3-6 shows the report options.
In Figure 3-7, observe the columns of data in the Performance List Extended report. This shows the default format of the report when a Report Form is not specified. It details performance-related information for each transaction, sorted by transaction ID.

Report tailoring

You can easily change the format of the Performance List Extended report by using a Report Form to include information to meet your specific reporting and analysis requirements. You can also tailor the sorting criteria by specifying up to three sort fields (ascending or descending) with (optionally) a limit on one. Many sample Report Forms of type LISTX are provided with CICS PA for this purpose. For more information about how to use Report Forms, see 2.10, “Tailoring report formats” on page 44.

Figure 3-8 shows how you can tailor the Performance List Extended report using a Report Form like the example in Figure 3-9. This example highlights bad response times for transactions that use DB2. This enables you to quickly analyze response time problems by identifying:

- The worst performing transactions
- The CICS internal and external resource that may have caused the problem
You can also write your Performance List Extended report (unsorted) to an extract data set. You do this by using the Export facility with a LIST or LISTX Report Form to define the record layout. When LISTX is used, the sort is ignored. The EOX marker in the Report Form defines the end of the extract record. There is no limit on the record length, so you can export all available fields or a selection. For more information, see 3.9.2, "Export extract" on page 101.
3.3.3 Performance Summary report

The Performance Summary report provides a summary of the CMF performance class records. Figure 3-10 shows the options for this report.

![Figure 3-10 Performance Summary report options]

**Default report format**

Observe the columns of data in the Performance Summary report in Figure 3-11. This shows the default format of the report when a Report Form is not specified. It summarizes by transaction ID. The Task Count (#Tasks) shows the number of performance class records processed during the reporting period.

![Figure 3-11 Performance Summary default report]

**Report tailoring**

Using a Report Form, you can easily change the format of the Performance Summary report to display the performance-related data in which you are interested. You can also tailor the sorting criteria by specifying up to three sort fields (ascending). Clock and Count fields can be...
summarized statistically by requesting any of these functions: Average, Minimum, Maximum, Standard Deviation, and Total. Many sample Report Forms of type SUMMARY are provided with CICS PA for this purpose. For more information about how to use Report Forms, see 2.10, "Tailoring report formats" on page 44.

Figure 3-12 shows how you can tailor the Performance Summary report using a Report Form as the one shown in Figure 3-13. This example shows the performance data summarized by Application Naming transaction ID within transaction ID by time of day. Note the following points:

- Transaction activity is summarized for each one-minute interval. The time interval defaults to one minute, but you can override this value and specify a time interval anywhere from one second to 24 hours (rounded down by CICS PA to align to the hour or day). This is an option on the report. It allows you to use the one Report Form for multiple reports with different report intervals.

- CICS Application Naming support allows you to monitor the performance of individual application transaction IDs (or programs) selected from a menu and run under one menu transaction ID. This is achieved by defining the field APPLTRAN (or APPLPROG) in the Report Form. This data is available under certain circumstances. For more information, see Chapter 14, “Application Naming support” on page 311.
Summary Export

You can also write your Performance Summary report to an extract data set. You can do this by using the Export facility with a SUMMARY Report Form to define the record layout and summarization criteria. The EOX marker in the Report Form defines the end of the extract record. There is no limit on the record length, so you can export all available fields or a selection. For more information, see 3.9.2, “Export extract” on page 101.

### 3.3.4 Performance Totals report

The Performance Totals report (Figure 3-14) provides a comprehensive analysis of the resource usage of your CICS system. You can use this report to gain a system-wide perspective of CICS system performance. Alternatively, you can use selection criteria to narrow the scope of the report. For example, you can specify, “Show me the resource usage for a particular group of transaction IDs or a single transaction ID or a single task number”.

![Figure 3-13  Performance Summary: SUMMARY Report Form](image)

#### Summary Export

You can also write your Performance Summary report to an extract data set. You can do this by using the Export facility with a SUMMARY Report Form to define the record layout and summarization criteria. The EOX marker in the Report Form defines the end of the extract record. There is no limit on the record length, so you can export all available fields or a selection. For more information, see 3.9.2, “Export extract” on page 101.
Chapter 3. Reports and extracts

Figure 3-14  Performance Totals report options

Report format

Figure 3-15 shows the Performance Totals report, which has four parts:

- **Overall CICS System Usage**: Reports CMF data about the CICS system as a whole:
  - CPU and dispatch times, broken down by TCB modes
  - Performance record and task counts

- **CPU and Dispatch statistics**: Provides a breakdown of the CPU, dispatch and suspend counts, and elapsed time. CPU time is broken down by each CICS Dispatcher TCB mode.

- **Resource Utilization statistics**: Each data field in the performance record is summarized into Total, Avg/Task and Max/Task, showing:
  - Count and time components for clock fields
  - Count values for count fields

- **User Field statistics**: Reports the statistics for the user fields (from any user-defined EMPs in the MCT) in the CMF performance class records.

---

**Table 3-15 Performance Totals report (Part 1 of 3)**
## Figure 3-16   Performance Totals report (Part 2 of 3)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Total</th>
<th>Avg/Task</th>
<th>Max/Task</th>
<th>Total</th>
<th>Avg/Task</th>
<th>Max/Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FCPUT</strong> File PUT requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>FCDELETE</strong> File DELETE requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>FCBROWSE</strong> File Browse requests</td>
<td>6556</td>
<td>10.2</td>
<td>1767</td>
<td>6556</td>
<td>10.2</td>
<td>1767</td>
</tr>
<tr>
<td><strong>FCADD</strong> File ADD requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TCC62OU2</strong> LU6.2 characters sent count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TCM62OU2</strong> LU6.2 messages sent count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TCC62IN2</strong> LU6.2 characters received count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TCM62IN2</strong> LU6.2 messages received count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TCCHROU1</strong> Terminal characters sent count</td>
<td>358311</td>
<td>556.4</td>
<td>1865</td>
<td>358311</td>
<td>556.4</td>
<td>1865</td>
</tr>
<tr>
<td><strong>TCMSGOU1</strong> Messages sent count</td>
<td>541</td>
<td>.0</td>
<td>2</td>
<td>541</td>
<td>.0</td>
<td>2</td>
</tr>
<tr>
<td><strong>TCCHRIN1</strong> Terminal characters received count</td>
<td>6996</td>
<td>10.9</td>
<td>225</td>
<td>6996</td>
<td>10.9</td>
<td>225</td>
</tr>
<tr>
<td><strong>TCMSGIN1</strong> Messages received count</td>
<td>537</td>
<td>.8</td>
<td>2</td>
<td>537</td>
<td>.8</td>
<td>2</td>
</tr>
<tr>
<td><strong>CFDTSYNC</strong> CF Data Table syncpoint wait time</td>
<td>538311</td>
<td>556.4</td>
<td>1865</td>
<td>538311</td>
<td>556.4</td>
<td>1865</td>
</tr>
<tr>
<td><strong>CMOD</strong> LU6.2 messages received count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>CC62OU2</strong> LU6.2 characters sent count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>CC62OU2</strong> LU6.2 messages sent count</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>FCADD</strong> File ADD requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>FCBROWSE</strong> File Browse requests</td>
<td>6556</td>
<td>10.2</td>
<td>1767</td>
<td>6556</td>
<td>10.2</td>
<td>1767</td>
</tr>
<tr>
<td><strong>FCDELETE</strong> File DELETE requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>FGET</strong> File GET requests</td>
<td>177</td>
<td>.3</td>
<td>137</td>
<td>177</td>
<td>.3</td>
<td>137</td>
</tr>
<tr>
<td><strong>FCPUT</strong> File PUT requests</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
</tr>
</tbody>
</table>
3.3.5 Wait Analysis report

The Wait Analysis report provides a breakdown of wait activity by transaction ID (or other ordering fields). You can see at a glance which CICS resources are causing your transactions to be suspended. This report can help you to quickly identify the possible source of a performance response time problem. Figure 3-18 shows the options for this report.

![Figure 3-18 Wait Analysis report options](image)

Report format

Figure 3-19 shows the Wait Analysis (Bottleneck) report. The Wait Analysis report has two sections.
The first section provides a summary of common performance metrics, including number of
Tasks, Response Time, Dispatch Time, CPU Time, Suspend Wait Time, Dispatch Wait Time,
RMI Elapsed Time, and RMI Suspend Time.

The ratio calculations on the right are particularly useful to see at a glance a possible
direction to look where the response/wait times are bad. In particular, observe that:

- CPU Time is shown as a percentage of Dispatch Time. This may indicate a possible lack
  of CPU.
- Dispatch Wait Time is shown as a percentage of Suspend Time. This may also indicate a
  lack of CPU or that another task is consuming the QR TCB. For example, if the Dispatch
  Wait Time was a significant amount of the Suspend Time, that indicates that the task is
  ready for dispatch but cannot for some reason.

The second section provides a detailed breakdown of suspend time by component, such as
dispatch wait, file wait, and so on. Components are reported in descending wait time order,
thereby ensuring that the primary cause of task wait is at the top of the list.

You can sort the report by up to three fields. The default is to summarize by transaction ID.

---

<table>
<thead>
<tr>
<th>Transactor</th>
<th>Start Time</th>
<th>Program</th>
<th>Interval</th>
<th>Summary Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td># Tasks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Response Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dispatch Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPU Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suspend Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dispatch Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Resource Manager Interface (RMI) elapsed time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Resource Manager Interface (RMI) suspend time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suspend Detail</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A Other Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DSPDELAY First dispatch wait time</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Transactor</th>
<th>Start Time</th>
<th>Program</th>
<th>Interval</th>
<th>Summary Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td># Tasks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Response Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dispatch Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPU Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suspend Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dispatch Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Resource Manager Interface (RMI) elapsed time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Resource Manager Interface (RMI) suspend time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suspend Detail</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A Other Wait Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DSPDELAY First dispatch wait time</td>
</tr>
</tbody>
</table>

---

Figure 3-19 Wait Analysis report
Recap report

The Wait Analysis report is always followed by the Wait Analysis Recap report to provide a breakdown of the CMF input data. Figure 3-20 shows the Recap report. It provides an overview of system-wide wait time. All CMF suspend components are reported in descending wait time order, ensuring that the primary cause of system-wide task wait is at the top of the list.

The Recap report shows all wait clocks, and even clocks that accumulated no wait time. This allows you to see at a glance:

- All the individual suspend component clocks
- Clocks that may be missing

---

Cross-System Work report

The Cross-System Work report correlates CMF performance class data by network unit of work (UOW) ID for a single CICS system or multiple CICS systems. Figure 3-21 shows the report options.

To run the report for multiple systems, define them to a Group. Groups enable you to connect systems together for consolidated reporting. This is especially useful for MRO, advanced program-to-program communication (APPC), or other systems that share workloads. For information about how to do this, see “Groups this system belongs to” on page 33.
Figure 3-21  Cross-System Work report options

**Default report format**

Observe the columns of data in the Cross-System Work report in Figure 3-22. This example shows the default format of the report when a Report Form is not specified. The default report includes only the performance class records that have the same network unit of work in multiple records (processing option 1).

Each line in the report is printed from a single CMF performance class record. Records that are part of the same network unit of work are printed sequentially in groups separated by a blank line.

The transaction Request Types are:

- **AP**: Application program request, including Distributed Program Link (DPL)
- **FS**: Function shipping request:
  - File Control (F)
  - Interval Control (I)
  - Transient Data (D)
  - Temporary Storage (S)
- **TR**: Transaction routing request for Terminal-Owning Region (TOR)
Chapter 3. Reports and extracts

Using a Report Form, you can easily change the format of the report to produce the Cross-System Work Extended report showing only the performance-related data in which you are interested. Many sample Report Forms of type LIST or LISTX are provided with CICS PA for this purpose. For this report, a LISTX Form is used in the same way as a LIST Form. That is, the fields and the order of the columns are used, but the sort sequence is ignored. For more information about how to use Report Forms, see 2.10, “Tailoring report formats” on page 44.

Figure 3-23 shows the Cross-System Work Extended report, produced by specifying a LIST or LISTX Report Form including dispatch statistics. The records are sorted by:

1. Network unit-of-work prefix (ascending)
2. Network unit-of-work suffix (ascending)
3. Syncpoint count concatenated with the task stop time (descending)
4. Generic APPLID (ascending)

Figure 3-23 shows the Cross-System Work Extended report, produced by specifying a LIST or LISTX Report Form including dispatch statistics. The records are sorted by:

- Network unit-of-work prefix (ascending)
- Network unit-of-work suffix (ascending)
- Syncpoint count concatenated with the task stop time (descending)
- Generic APPLID (ascending)
Cross-System Work extract
You can also request a Cross-System Work extract. This extract combines CMF performance class records that belong to the same network unit of work into a single CMF-format record to provide a complete view of a transaction’s CICS resource usage. You can then use the extract as input to CICS PA to produce any of the reports and extracts. For more information, see 3.9.1, “Cross-System Work extract” on page 99.

3.3.7 Transaction Group report
The Transaction Group report is used to help you understand the correlation of the performance class records that are attached in a CICS assigned transaction group.

The Transaction Group ID (TRNGRPID) is assigned internally by CICS at transaction attach time. CICS PA uses this ID to correlate the transactions belonging to the same work request, such as the CWXN (Web Attach) and CWBA (Alias transaction). Figure 3-24 shows the report options.

Figure 3-24 Transaction Group report options

Report format
Figure 3-25 shows the format of the Transaction Group report. The Origin field can help you understand the flow of transactions through a CICS system when applied to transaction requests that originate through:

- CICS Web Support (CWS)
- Internet Inter-ORB Protocol (IIOP)
- External Call Interface (ECI) over TCP/IP
- 3270 Bridge “two-task model”

The detailed report is followed by a Summary report that summarizes and groups the transactions by their origin.
3.3.8 BTS report

The BTS report provides a detailed report of the transactions performed by the same or different CICS systems on behalf of a single CICS BTS process. Figure 3-26 shows the report options.

![BTS report options](image)

**Figure 3-26  BTS report options**
Report format

Figure 3-27 shows the format of the BTS report. The BTS report is similar to the Cross-System Work and Transaction Group reports in that it is a detailed report. However, this report shows the correlation of the transactions performed by the same or different CICS systems on behalf of a single CICS BTS process (root activity ID).

The records are sorted by:

- BTS Process ID (Root Activity ID)
- Transaction Sequence Number
- Transaction Stop Time (ascending order)

<table>
<thead>
<tr>
<th>Tran SC</th>
<th>TranType</th>
<th>Process Name</th>
<th>Activity Name</th>
<th>Pro/Act</th>
<th>Cont’r</th>
<th>Event</th>
<th>R</th>
<th>Task T</th>
<th>Stop Time</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAL1</td>
<td>TP</td>
<td>U</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>211 T11:18:25.27</td>
<td>.1222</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAL1</td>
<td>TP</td>
<td>U</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>239 T11:19:18.33</td>
<td>.1835</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAY1</td>
<td>TP</td>
<td>U</td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>294 T11:19:42.20</td>
<td>.1390</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAY1</td>
<td>TP</td>
<td>U</td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>305 T11:19:57.64</td>
<td>.0747</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RED1</td>
<td>U</td>
<td>R SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>176 T11:17:32.05</td>
<td>.5333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STDC</td>
<td>U</td>
<td>R SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>177 T11:17:32.05</td>
<td>.5145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALE</td>
<td>U</td>
<td>R SALES11111</td>
<td>ORDER</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>175 T11:17:32.05</td>
<td>.5675</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INV1</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>178 T11:17:32.09</td>
<td>.0159</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEL1</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>179 T11:17:33.29</td>
<td>1.2323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALE</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>180 T11:17:33.31</td>
<td>1.2198</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALE</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>183 T11:17:33.37</td>
<td>.0800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALE</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>184 T11:17:33.42</td>
<td>.0519</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALE</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>186 T11:17:38.65</td>
<td>.0566</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REM1</td>
<td>U</td>
<td>SALES11111</td>
<td>ORDER</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>187 T11:17:38.68</td>
<td>.0243</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-27  BTS report

3.3.9 Workload Activity report

The Workload Activity report provides a detailed list, summary, or list and summary of the segments of work (transactions) performed by the same or different CICS systems through transaction routing, function shipping, or distributed transaction processing on behalf of a single network unit of work. Figure 3-28 shows the report options.
Chapter 3. Reports and extracts

### Figure 3-28  Workload Activity report options

**Report format**

Figure 3-29 shows the Workload Activity report. This report highlights the MVS Workload Manager (WLM) service class and report class, and WLM reporting and completion phase (BTE or EXE) used for each transaction.

The Workload Activity Summary report summarizes response time by WLM service and report classes. The statistics it provides are Average, Standard Deviation, \( nn\% \) Peak, and Maximum.

---

**Figure 3-29  Workload Activity report**

<table>
<thead>
<tr>
<th>Service Class</th>
<th>APPLID</th>
<th>Phase</th>
<th>#Tasks</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Other</em></td>
<td>STMTIRA1</td>
<td>BTE</td>
<td>105</td>
<td>0.0009 0.0007 0.0018 0.0072</td>
</tr>
<tr>
<td></td>
<td>STMTIRA2</td>
<td>BTE</td>
<td>174</td>
<td>0.0008 0.0002 0.0010 0.0019</td>
</tr>
<tr>
<td></td>
<td>STMTIRT0</td>
<td>BTE</td>
<td>589</td>
<td>1.1839 8.8946 12.5868 135.009</td>
</tr>
<tr>
<td></td>
<td>STMTIRT1</td>
<td>BTE</td>
<td>551</td>
<td>1.7020 9.7902 14.2531 133.831</td>
</tr>
<tr>
<td></td>
<td>STMTIRT2</td>
<td>BTE</td>
<td>570</td>
<td>2.1656 13.4634 19.4257 176.251</td>
</tr>
<tr>
<td></td>
<td>STMTIRT3</td>
<td>BTE</td>
<td>570</td>
<td>1.4052 9.6969 13.8366 149.703</td>
</tr>
<tr>
<td></td>
<td>STMTIRT4</td>
<td>BTE</td>
<td>570</td>
<td>1.4566 8.0846 11.8503 135.869</td>
</tr>
<tr>
<td></td>
<td>STMTIRT5</td>
<td>BTE</td>
<td>570</td>
<td>2.3631 14.1819 20.5443 179.756</td>
</tr>
</tbody>
</table>
3.4 Exception reports

The Exception reports are:

▶ Exception List
▶ Exception Summary

3.4.1 Exception List report

The Exception List report provides detailed analysis of the exception class records collected by the CICS Monitoring Facility (CMF). Figure 3-30 shows the report options.

![Exception List report options](image)

**Report format**

Figure 3-31 shows an example of the Exception List report. The Exception List report provides two types of information:

▶ The cause of the exception condition
▶ The information necessary to relate this record to the performance class record on the Performance List report

![Exception List report](image)
3.4.2 Exception Summary report

The Exception Summary report summarizes the exception class records collected by the CICS Monitoring Facility (CMF). Figure 3-32 shows the report options.

**Figure 3-32  Exception Summary report options**

**Report format**

Figure 3-33 shows the Exception Summary report. The exception class records are summarized by transaction ID. The report provides the total number of exceptions for each transaction, according to:

- Auxiliary temporary storage Virtual Storage Access Method (VSAM) buffer and string wait conditions
- VSAM LSRPOOL buffer and string wait conditions
- VSAM file string wait conditions
- Temporary storage wait conditions
- Main storage wait conditions
- Coupling facility data table pool wait conditions

**Figure 3-33  Exception Summary report**

3.5 Transaction Resource Usage reports

The Transaction Resource Usage reports are:

- File Usage Summary report
- The Temporary Storage Usage Summary report
- The Transaction Resource Usage List report
### 3.5.1 File Usage Summary report

The Transaction File Usage Summary report summarizes file usage by transaction ID. For each transaction ID, it gives transaction information and file control statistics followed by a breakdown of file usage for each file used. Figure 3-34 shows the report options.

![Figure 3-34 Transaction Resource Usage: File Usage Summary report options](image)

**Command**

```plaintext
REDBOOK - File Usage Summary Report
Command ==> __________________________________________________________________________
System Selection:                       Report Output:
APPLID . . SCSCPAA5 +                  DDname . . . . . . . . . FILE0001
Image . . ________ +                    Print Lines per Page . . ___ (1-255)
Group . . ________ +                    
Summary Reports Required:
/ Transaction File Usage
/ File Usage
/ Break down by Transaction ID
/ Include Transaction Totals
Report Format:
Title . . __________________________________________________________________________
Selection Criteria:
| Performance |
```

**Report format**

Figure 3-35 shows an example of the File Usage Summary report. This report summarizes file activity. For each file, it gives a breakdown of file usage by transaction ID.

![Figure 3-35 Transaction Resource Usage: File Usage Summary report](image)
3.5.2 Temporary Storage Usage Summary report

The Transaction Temporary Storage Usage Summary report summarizes temporary storage queue usage by transaction ID. For each transaction ID, it gives transaction information and temporary storage statistics followed by a breakdown of Tsqname usage for each temporary storage queue used.

The Temporary Storage Usage Summary report summarizes Tsqueue activity. For each Tsqueue, it gives a breakdown of Temporary Storage Queue usage by transaction ID. Figure 3-36 shows the report options.

```
Command ===> _________________________________________________________________
System Selection:                       Report Output:
  APPLID . . SCSCPAA5  +                  DDname . . . . . . . . . FILE0001
  Image . . ________  +                  Print Lines per Page . . ___ (1-255)
  Group . . ________  +

Summary Reports Required:
/  Transaction Temporary Storage Usage
/  Temporary Storage Usage
/  Break down by Transaction ID
/  Include Transaction Totals

Report Format:
  Title . . ________________________________________________________________
  ________________________________________________________________

Selection Criteria:
  Performance
```

**Figure 3-36** Transaction Resource Usage: Temporary Storage Summary report options

**Report format**

Figure 3-37 shows the Temporary Storage Usage Summary report.

```
VIR3MD CICS Performance Analyzer
Temporary Storage Usage Summary

TEMP0001 Printed at 11:00:52  7/26/2003  Data from 07:30:47  5/29/2003 to 08:35:48  5/29/2003  APPLID CICSPA1  Page 1

<table>
<thead>
<tr>
<th>TSQueue</th>
<th>Tran</th>
<th>#Tasks</th>
<th>Elapse Avg</th>
<th>Put_AUX</th>
<th>Put_Main</th>
<th>Total</th>
<th>Shr_TS</th>
<th>Get</th>
<th>Put_AUX</th>
<th>Put_Main</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS_QUEUE1</td>
<td>CEDA</td>
<td>9</td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0106</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0104</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Count</td>
<td>Avg</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>56</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>3</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>17</td>
<td>112</td>
</tr>
<tr>
<td>CSSY</td>
<td>4</td>
<td></td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Count</td>
<td>Avg</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>56</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>3</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>17</td>
<td>112</td>
</tr>
<tr>
<td>Totl</td>
<td>13</td>
<td></td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>.0104</td>
<td>.0000</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>.0139</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Count</td>
<td>Avg</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>56</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max</td>
<td>3</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>17</td>
<td>112</td>
</tr>
</tbody>
</table>
```

**Figure 3-37** Transaction Resource Usage: Temporary Storage Summary report
3.5.3 Transaction Resource Usage List report

The Transaction Resource Usage List report provides a list of all transaction resource class records in the sequence that they appear in the SMF file. It gives transaction information, detailing their individual file and temporary storage queue usage. Figure 3-38 shows the report options.

![Figure 3-38: Transaction Resource Usage List report options](Image)

**Report format**

Figure 3-39 shows an example of the Transaction Resource Usage List report.
### 3.6 Subsystem reports

The Subsystem reports are:

- DB2 report
- The WebSphere MQ report

#### 3.6.1 DB2 report

The CICS PA DB2 report combines the CICS CMF performance class records (SMF 110) with the DB2 Accounting records (SMF 101) that belong to the same network unit of work, including some DB2 activity. It can provide a detailed, summary, or detailed summary report showing DB2 usage for your CICS systems.
The DB2 reports are:

- List
- Summary (short or long)
- Recap (record processing statistics)

To produce the DB2 reports, you need to accumulate DB2 accounting statistics (SMF 101 records) and define your CICS-DB2 resources with ACCOUNTREC(TASK) or ACCOUNTREC(UOW). CICS PA V1R3 supports the DB2 accounting statistics data from DB2 Version 5, Version 6, and Version 7.

You can use the information provided in the CICS PA DB2 reports to assist in further analysis using DB2 performance reporting tools such as the DB2 Performance Monitor (DB2 PM).

The CICS PA DB2 List report is most effective when used in conjunction with the CICS PA Cross-System Work report. Figure 3-40 shows the report options.

---

**Report format**

Figure 3-41 shows the DB2 List report. This report provides a detailed list by transaction of all network units of work with DB2 activity. Records that are part of the same network unit of work are printed sequentially in groups with a blank line separator. A data line (column format) is presented for each CMF performance class record. A block of data lines (row format) is presented for each associated DB2 Accounting record.

The DB2 Long Summary report summarizes DB2 activity by transaction and program (CMF performance records), and SSID and Plan name (DB2 accounting records) within APPLID. Average and maximum values are reported for each. This report represents a subset of the total data presented in the DB2 List report. It includes DB2 data that can be matched within network unit of work to a single task, or to multiple tasks for the same transaction and program.
The DB2 Short Summary report is an abridged version of the Long Summary report. It provides averages only (no maximums). Both the CMF performance and DB2 accounting record details are presented in column format.

The DB2 Recap report is always produced at the end to provide an analysis of the CICS CMF performance class (SMF 110) and the DB2 Accounting (SMF 101) records processed.

---

### Figure 3-41  DB2 List report

#### 3.6.2  WebSphere MQ report

The CICS PA MQ reports use the WebSphere MQ accounting data (SMF 116 records) to provide a detailed performance analysis of the CICS transactions that access an MQ queue manager. **CICS PA Version 1 Release 3** supports the WebSphere MQ accounting statistics data from MQSeries for OS/390 Version 5.2, IBM WebSphere MQ for z/OS Version 5.3, and IBM WebSphere MQ for z/OS Version 5.3.1.

The CICS PA MQ List reports provide a detailed trace of the WebSphere MQ accounting records, reporting the comprehensive performance data contained in the **Class 1** and **Class 3** records:

- **Class 1 (Subtype 0)**: Message manager accounting records, record how much CPU was spent processing WebSphere MQ API calls and the number of MQGET and MQPUT calls.
- **Class 3 (Subtypes 1 and 2)**: Accounting data for each task, at thread and queue level.

The MQ Summary reports provide, summarized by either CICS transaction ID, MQ queue name, or both, an analysis of the MQ system and queue resources used and the transactions they service.
Figure 3-42 shows the WebSphere MQ report options.

**Report format**

The WebSphere MQ SupportPac MP1B: MQSeries for OS/390 V5.2 - Interpreting accounting and statistics data provides information about the use and interpretation of the accounting and statistics available in MQSeries for OS/390 Version 5.2 (and later). It also provides information about the layout of the SMF records and suggests ways to analyze the data.

Figure 3-43 shows the WebSphere MQ Class 1 List report.
Figure 3-44 shows the WebSphere MQ Class 1 Summary report.

<table>
<thead>
<tr>
<th>SSID</th>
<th>APPLID</th>
<th>TRAN</th>
<th>Count</th>
<th>CPU</th>
<th>Calls</th>
<th>&lt;=99</th>
<th>&lt;=999</th>
<th>&lt;=9999</th>
<th>&gt;=10000</th>
<th>&lt;=99</th>
<th>&lt;=999</th>
<th>&lt;=9999</th>
<th>&gt;=10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQMD</td>
<td>CICS53A1</td>
<td>CKCN</td>
<td>1</td>
<td>0.000747</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MQMD</td>
<td>CICS53A1</td>
<td>CKTI</td>
<td>1</td>
<td>0.001541</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MQMD</td>
<td>CICS53A1</td>
<td>MQAI</td>
<td>1</td>
<td>0.064342</td>
<td>60.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>60.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 3-44  WebSphere MQ Class 1 Summary report

Figure 3-45 shows the WebSphere MQ Class 3 List report.

<table>
<thead>
<tr>
<th>SSID: MQMD</th>
<th>APPLID: CICS53A1</th>
<th>Tran: MQAI</th>
<th>Task: 41</th>
<th>UserID: CICSUSER</th>
<th>NetName: N/A</th>
<th>UOWID: N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>Total Calls</td>
<td>1</td>
<td>Avg Elapsed</td>
<td>0.018721</td>
<td>Avg CPU</td>
<td>0.000258</td>
</tr>
<tr>
<td></td>
<td>#Old Pages</td>
<td>120</td>
<td>#New Pages</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queue:</td>
<td>CPPX.MQS520.TEST.TEMPQUEUE.060</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QType:</td>
<td>LOCAL</td>
<td></td>
<td>IType: NONE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>Elapsed</td>
<td>0.000332</td>
<td>Susp Elp</td>
<td>0.000327</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
<td></td>
<td>JnlWrt Elp</td>
<td>PS Req's</td>
<td>PS Rd Elp</td>
<td>Expired</td>
</tr>
<tr>
<td>PUT</td>
<td></td>
<td>0.000567</td>
<td>0.000560</td>
<td>0.0000000</td>
<td>0.0</td>
<td>0.000000</td>
</tr>
<tr>
<td>PUT Total Bytes</td>
<td>10</td>
<td>#PUT w/Data</td>
<td>1</td>
<td>Min Msg Size</td>
<td>10</td>
<td>Max Msg Size</td>
</tr>
<tr>
<td>Queue:</td>
<td>CPPX.MQS520.TEST.TEMPQUEUE.059</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QType:</td>
<td>LOCAL</td>
<td></td>
<td>IType: NONE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>Elapsed</td>
<td>0.000332</td>
<td>Susp Elp</td>
<td>0.000327</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
<td></td>
<td>JnlWrt Elp</td>
<td>PS Req's</td>
<td>PS Rd Elp</td>
<td>Expired</td>
</tr>
<tr>
<td>PUT</td>
<td></td>
<td>0.000567</td>
<td>0.000560</td>
<td>0.0000000</td>
<td>0.0</td>
<td>0.000000</td>
</tr>
<tr>
<td>PUT Total Bytes</td>
<td>10</td>
<td>#PUT w/Data</td>
<td>1</td>
<td>Min Msg Size</td>
<td>10</td>
<td>Max Msg Size</td>
</tr>
</tbody>
</table>

Figure 3-45  WebSphere MQ Class 3 List report
3.7 System reports

The System report category includes the System Logger report.

3.7.1 System Logger report

The System Logger reports process the System Logger (SMF 88) records to provide information about the System Logger log streams and coupling facility structures that are used by CICS Transaction Server for logging, recovery and backout operations.

The CICS PA System Logger reports, when used in conjunction with the CICS Logger reports produced by the standard CICS statistics reporting utilities, provide a comprehensive analysis of the logstream activity for all your CICS systems. They also provide a more extensive and flexible performance reporting solution than the IXGRPT1 sample program.

Figure 3-47 shows the report options.
Figure 3-47  System Logger report options

You can request a List report, a Summary report, or both. The System Logger List report shows information about logstream writes, deletes, and events (Subtype 1), as well as structure alter events (Subtype 11) for each SMF recording interval. Structure alter events apply to structures, not individual log streams. They are reported with a logstream name of *ALTER*. The report is sorted either on logstream name or structure name.

The System Logger (SMF 88) records can be filtered by logstream, structure, or both name patterns. The masking characters % and * are also supported.

The System Logger Summary report summarizes logstream and structure statistics so that you can measure logger performance over a longer period of time.
### Report format

Figure 3-48 shows the System Logger - Logstream Summary report.

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>MVSID</th>
<th>Structure name</th>
<th>First interval start</th>
<th>Last interval stop</th>
<th>Total Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>IYOT1.DFHLOG</td>
<td>SYSD</td>
<td>LOG_JG_20M</td>
<td>23:00:00.00  1/05/2002</td>
<td>23:46:22.38  1/05/2002</td>
<td>00:00:46:22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IXGWRITES</th>
<th>Bytes写入</th>
<th>Count</th>
<th>Count</th>
<th>Bytes写入</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
<th>银行</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Total</td>
<td>Average</td>
<td>Interim</td>
<td>DASD</td>
<td>DASD</td>
<td>Offload</td>
<td>w/o DASD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>628147</td>
<td>172706K</td>
<td>275</td>
<td>301535K</td>
<td>216244</td>
<td>46717</td>
<td>59484K</td>
<td>128572K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate(次/秒)</td>
<td>225</td>
<td>62080</td>
<td>108388</td>
<td>77</td>
<td>168</td>
<td>21362</td>
<td>46216</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>4</td>
<td>4292</td>
<td>106388</td>
<td>77</td>
<td>168</td>
<td>21362</td>
<td>46216</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>94200</td>
<td>25898K</td>
<td>45218K</td>
<td>32740</td>
<td>71810</td>
<td>9004730</td>
<td>19739K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENTS</th>
<th>Demand</th>
<th>Demand</th>
<th>Demand</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offloads</td>
<td>314</td>
<td>0</td>
<td>78</td>
<td>0</td>
</tr>
<tr>
<td>Rate(次/秒)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Minimum</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>116</td>
</tr>
<tr>
<td>Maximum</td>
<td>48</td>
<td>0</td>
<td>12</td>
<td>1427</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENTS</th>
<th>DASD写入</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type1</td>
<td>Type2</td>
</tr>
<tr>
<td>Rebuilds Init'd</td>
<td>315</td>
</tr>
<tr>
<td>Rebuilds Compl't</td>
<td></td>
</tr>
<tr>
<td>Type1</td>
<td>Type2</td>
</tr>
<tr>
<td>612865</td>
<td>15277</td>
</tr>
<tr>
<td>Rate(次/秒)</td>
<td>220</td>
</tr>
<tr>
<td>Minimum</td>
<td>4</td>
</tr>
<tr>
<td>Maximum</td>
<td>91995</td>
</tr>
</tbody>
</table>
Figure 3-49 shows the System Logger - Structure Summary report.

### 3.8 Performance Graph reports

The Performance Graph reports are:

- Transaction Rate Graph report
- Transaction Response Time Graph report

#### 3.8.1 Transaction Rate Graph report

The Transaction Rate Graph report shows, over the requested time interval, the average response time and the number of completed transactions. Figure 3-50 shows the report options.
Figure 3-50  Transaction Rate Graph report options

**Report format**

Figure 3-51 shows the Transaction Rate Graph report.

Figure 3-51  Transaction Rate Graph report

### 3.8.2 Transaction Response Time Graph report

The Transaction Response Time Graph report shows the average and maximum response time. Figure 3-52 shows the report options.
### 3.9 Performance extracts

The Performance extracts are:

- Cross-System Work extract
- Export extract
- Record Selection extract

#### 3.9.1 Cross-System Work extract

The Cross-System Work extract consolidates the CMF performance class records that belong to the same network unit of work into a single record in CMF performance record format. You can then use the extract data set as input to other CICS PA reports or extracts such as a Performance List report or a Performance Data extract. Figure 3-54 shows the extract options.

---

**Figure 3-52 Transaction Response Time Graph report options**

**Report format**

Figure 3-53 shows the Transaction Response Time Graph report.

---

**Figure 3-53 Transaction Response Time Graph report**
All CMF fields are available for inclusion in the extract. In addition, you can specify which user fields you want to include.

To run the report for multiple systems, define them to a group. For information about how to do this, see “Groups this system belongs to” on page 33.

**Extract record format**

Figure 3-55 shows an example of the Cross-System Work extract record:

- The extract records are written for the specified APPLID/MVS ID. The default is MULTIPLE/CICS.
- Transactions are identified by the originating task.
- Counters and elapsed times are combined to provide a complete view of a transaction's CICS resource usage.
Recap report
A Recap report is always produced to provide the total record count in the extract data set.

Using CICS PA to process the extract
You can input the Cross-System Work extract data set into CICS PA for further analysis. Figure 3-56 shows an example of the System Definitions screen after you define the Cross-System Work APPLID (MULTIPLE) and MVS ID (CICS) to CICS PA and associate the extract file with the APPLID. Then you can define report forms and report sets to run on this APPLID.

3.9.2 Export extract
The Export extract is created as a delimited text file for the purpose of importing the CMF performance class data into PC spreadsheet or database tools for further analysis and reporting. CICS PA supplies the column headings (if requested). The fields are separated by
a delimiter character of your choosing (the default is the semi-colon (;)). Figure 3-57 shows the extract options.

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>System Selection:</th>
<th>Extract Recap:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>APPLID . . SCSCPAA5 +</td>
<td>DDname . . EXPT0001</td>
</tr>
<tr>
<td></td>
<td>Image . . +</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group . . +</td>
<td></td>
</tr>
<tr>
<td>Output Data Set . . 'SCSCPAA5.EXPORT'</td>
<td>Disposition . . 1 1. OLD 2. MOD (If cataloged)</td>
<td></td>
</tr>
<tr>
<td>Extract Format:</td>
<td>Enter &quot;/&quot; to select option</td>
<td></td>
</tr>
<tr>
<td>Form . . +</td>
<td>Include Field Labels</td>
<td></td>
</tr>
<tr>
<td>Delimiter . ;</td>
<td>Numeric Fields in Float format</td>
<td></td>
</tr>
<tr>
<td>Selection Criteria:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_ Performance *</td>
<td>Summary Processing Options:</td>
<td></td>
</tr>
<tr>
<td>LIST, LISTX or SUMMARY Report Form</td>
<td>Time Interval 00:01:00 (hh:mm:ss)</td>
<td></td>
</tr>
</tbody>
</table>

**Default extract record format**

When a Report Form is not specified, the default Export record format contains these fields:

- **APPLID**: Generic APPLID
- **Tran**: Transaction ID
- **Term**: Terminal ID
- **Userid**: User ID
- **Taskno**: Transaction sequence number
- **Stop Date**: Transaction stop date (yyyy-mm-dd)
- **Stop Time**: Transaction stop time (hh:mm:ss.thm)
- **Response**: Transaction response time
- **Clocks**: All 65 clocks as defined by CICS Transaction Server for z/OS, Version 2.2

Figure 3-58 shows an example of the first part of the default record layout. Note that the field labels (column headings) are included in this extract. Field labels are optional.
Extract record tailoring (LIST, LISTX)

You can specify a LIST or LISTX Report Form to create an Export record in the same format as the corresponding Performance List report (see Figure 3-4 on page 66) or Performance List Extended report (see Figure 3-8 on page 69). Many sample Report Forms of type LIST or LISTX are provided with CICS PA for this purpose. Note that when you use a LISTX Form for the Export, it is used like a LIST Form. That is, the fields and the order of the columns are used, but the sort sequence is ignored. For more information about how to use Report Forms, see 2.10, “Tailoring report formats” on page 44.

Extract record tailoring (SUMMARY)

You can specify a SUMMARY Report Form to create an Export record in the same format as the corresponding Performance Summary report (see Figure 3-12 on page 71). Many sample Report Forms of type SUMMARY are provided with CICS PA for this purpose. For more information about how to use Report Forms, see 2.10, “Tailoring report formats” on page 44.

Recap report

A Recap report is always produced to give the total record count in the extract data set.

Processing the extract using different tools

The Export data set is a delimited text file. You can analyze this file further by using a program, such as DB2, or PC tools, such as Lotus 1-2-3 or Lotus Approach. Figure 3-59 is an example of a graph produced from the Summary Export data.

For more examples and descriptions of how to produce such graphs, see Chapter 4, “Processing extracts” on page 107.
3.9.3 Record Selection extract

The Record Selection extract is a facility that allows you to create a smaller extract file containing only the CMF performance records (and optionally, the DB2 and WebSphere MQ accounting records) that are of interest to you. It is used to filter large SMF files. You can then use these files as input to CICS PA, for more efficient reporting and analysis. Figure 3-60 shows the extract options.

```
Command ===>

System Selection:
  CICS APPLID . . SCSCPAA5 + Image . . ______ + Group . . ______ +
  DB2 SSID . . DB2P + Image . . ______ + Group . . ______ +
  MQ SSID . . . . ____ + Image . . ______ + Group . . ______ +

Extract Recap:
  DDname . . RSEL0001

Output Data Set:
  Data Set Name . . 'SCSCPAA5.DB2P.RECSEL'
  Disposition . . _ 1. OLD 2. MOD (If cataloged)

Selection Criteria:
  _ Performance
```

**Figure 3-60  Record Selection Extract options**

**Extract record format**

The extract file contains CMF performance records (SMF 110) and, if requested, DB2 Accounting records (SMF 101).
Recap report
A Recap report is always produced at the end of extract processing. Figure 3-61 shows how the Recap report summarizes the results of the extract processing.

Figure 3-61 Record Selection extract (Recap report)

3.10 Popular mix

Do not be daunted. You will soon learn what mix of reports works best for you for your ongoing monitoring and tuning efforts and for your management reporting.

A suggestion for monthly reporting and beyond is to use the Historical Database facility to maintain a history of CMF performance data for reporting or export to DB2 tables. See Chapter 19, “Historical Database” on page 415.

For daily reporting, we recommend:

- Performance Summary (1): Use a SUMMARY Report Form to show transaction count, response time, CPU, and so on, summarized by transaction within APPLID.
- Performance Summary (2): The same as Performance Summary (1), but summarized by transaction within a group (production, test, etc).
- Performance List Extended: Showing the top 20 poor response times, and so on, by transaction within APPLID.
- Wait Analysis.
- Exception Summary.
- DB2 or WebSphere MQ Summary.
- Logstream Summary.
- Performance Export: Process the extract using PC tools to produce graphs.

For weekly reporting, we recommend:

- Performance Summary: As for daily reporting; summarized by transaction within a group.
- Performance Totals: By Group.
- DB2 or WebSphere MQ Summary.
- Logstream Summary.
- Performance Export. Process the extract using DB2 or PC tools to produce graphs.

For monthly reporting, we recommend:

- Performance Summary: As for daily reporting; summarized by transaction within a group
- Performance Totals: By Group
- DB2 or WebSphere MQ Summary
Chapter 4. Processing extracts

The CICS Performance Analyzer (PA) Export facility produces delimited text files of CICS Monitoring Facility (CMF) Performance and Transaction Resource Class data extracted from SMF data sets. The extract files are suitable for analysis by external programs such as DB2, or PC spreadsheet and graphing tools such as Lotus 1-2-3, Microsoft® Excel and Microsoft Access.

This chapter explains, by example, how to:

- Use the CICS PA Export facility to produce extract data sets in various formats
- Use the following programs and tools to process the extract data sets:
  - DB2
  - Lotus 1-2-3
  - Microsoft Excel
  - Microsoft Access

Methods such as these can enhance your understanding and interpretation of the data, facilitate comparisons, assist your analysis of trends, peaks and throughputs, help to isolate problems, and generally support your decision-making about the ongoing tuning and management of your CICS systems.
4.1 Processing extracts with DB2

CICS Performance Analyzer 1.3 introduced an easy way to generate data from SMF records in a form that is suitable to be used as input for a DB2 load utility and to be imported into a DB2 table. Furthermore CICS PA generates the Data Definition Language (DDL) statements to create the required DB2 database, tablespace, table, and an index. Also the load job is generated by CICS PA.

This function is part of the Historical Database and is demonstrated in 19.4.5, “Export HDB data sets to DB2” on page 441.

4.2 Processing extracts with Lotus 1-2-3

A chart is an effective way to illustrate your performance data after it is placed in a spreadsheet. It can make relationships among numbers easy to see because it turns numbers into shapes. This section shows how to create a chart based on performance data that was extracted using the CICS Performance Analyzer.

4.2.1 Exporting performance extracts

From the Primary Option Menu, select option 2 (Report Sets). The Report Sets screen shows a list of Report Sets that were already created. Refer to 2.7.2, “Report Sets” on page 36, for a description of how to create a Report Set.

We typed line action S next to Report Set VSAMSUM. This displays the Edit Report Set screen. Report Set VSAMSUM was created when we ran the VSAM performance scenario described in Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139.

On the Edit Report Set screen, we checked that no active reports were selected. Active reports are indicated by a yes after the report name. You can use line action D to deactivate the report. After that, we selected the item Export from the Extracts group. The Exports screen appears and displays a list of selections from our previous extract exports. Figure 4-1 shows the Export screen containing a list of systems to select from.

```
Figure 4-1   Performance extract export
```

```
VSAMSUM - Exports                      Command ====>  Row 1 from 3  Scroll ====> CSR_

----- System Selection ----- Selection
/ Exc APPLID + Image + Group + Recap Form + Criteria
SCSCPAA1 SC66 EXPT0001 VSAMEL YES

Output Data Set . . LOTUS123

SCSCPAA1 SC66 EXPT0001 VSAMSUM YES

Output Data Set . . MSACCESS

SCSCPAA1 SC66 EXPT0001 VSAM YES

Output Data Set . . MSEXCEL
```
We selected system SCSCPAA1 using output data set LOTUS123. This opens an Export screen that allows you to edit export detail information. If you do not see a screen like the one shown in Figure 4-1, then this may be the first performance extract export that bypasses the list of previous exports and displays the screen shown in Figure 4-2 directly.

We updated the following information on the Export detail screen:

- CICS APPLID SCSCPAA1
- Extract data set LOTUS123
- Report form VSAMSUM

![VSAMSUM - Export](image)

**Figure 4-2  Extract details**

CICS system definition SCSCPAA1 must specify a valid SMF data set name that contains some SMF records to produce a performance extract. Refer to “SMF Files for this system” on page 33 for a description about how to define SMF data sets on system definitions.

We used Report Form VSAMSUM. It was created during the VSAM performance scenario project described in Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139. To demonstrate how to create charts, you can use any Report Form that has time or count fields.
We pressed F3 until we returned to the Report Set screen. We typed RUN to create and submit the JCL to process the performance extract export. When the job has completed, the extracted data was stored to data set CICSL4.LOTUS123 (Figure 4-3).

![Figure 4-3 Performance extract](image)

We used file transfer to download the performance extract to a Windows 2000 workstation. The file was stored to directory c:\reports\lotus1-2-3\lotus_chart.txt. Note that the title line of the extracted data is not aligned with the columns. Lotus 1-2-3 aligns the delimiters during the import process.

### 4.2.2 Importing extracted data to Lotus 1-2-3

To import the extracted performance data to Lotus 1-2-3, we clicked Start->Lotus Smart Suite->Lotus 1-2-3. When the Lotus 1-2-3 main window opened, we clicked File->Open. A window opens that allows the navigation to the directory which contains our text file. We double-clicked our file, lotus_chart.txt.

The Text File Options window (Figure 4-4) opens. We selected the Start a new column at each radio button and selected Semicolon from the list in the text box. We had to do that since we used a semicolon as delimiter character when we exported the performance extract. We clicked OK to continue.

![Figure 4-4 Text File Options window](image)

Lotus 1-2-3 imported our performance data and displayed a data sheet (Figure 4-5) that corresponds with the columns of our performance extract. We create a chart that visually illustrates a comparison of average response time and maximum response time per transaction of our VSAM workload.
Lotus 1-2-3 provides a convenient way to create basic charts. You can set up a range so that it contains all the elements you need to create a basic chart.

The range that we wanted to use has text and numbers arranged as in A1-C11 as illustrated in Figure 4-5. Lotus 1-2-3 plots the chart based on range A1-C11 by column. Lotus 1-2-3 automatically creates a bar chart if columns A1-C11 are selected as the data range.

It is also possible to insert two additional rows, containing the title and subtitle of the chart, in front of the data sheet. The new data range is then A1-C13. We decided that it is more convenient to alter the chart afterward rather than to modify the data sheet. Figure 4-6 shows how we selected the data range from which we wanted to create the chart.

Figure 4-5  Lotus 1-2-3 data sheet

Figure 4-6  Creating a chart
The data range comprises A1 to C11, so we have more rows than columns. This influences the way the chart is built automatically. Lotus 1-2-3 builds charts automatically in the following ways:

- More columns than rows: By interpreting each row of values as a separate series, Lotus 1-2-3 uses the leftmost entry in each row as a legend label and the top entry in each column as an axis label.
- More rows than columns: By interpreting each column of values as a separate series, Lotus 1-2-3 uses the top entry in each column as a legend label and the leftmost entry in each row as an axis label.
- Equal rows and columns: This situation is handled in the same way as when there are more rows than columns.

We clicked **Create->Chart** and dropped the chart icon into the selected data range. The chart was built automatically as shown in Figure 4-7. The layout of the chart is still very basic, but can easily be improved. So far, we have not specified a specific chart type, axes and grids, chart style, or chart options. For simplicity, we only showed the basic technique to create charts using Lotus 1-2-3.

![Lotus 1-2-3 chart](image)

**Figure 4-7  Lotus 1-2-3 chart**

### 4.3 Processing extracts with Microsoft Excel

Microsoft Excel allows the creation of charts by using a chart wizard. You can use any existing data sheet to start the chart wizard to create charts from selected columns.

#### 4.3.1 Exporting a performance extract

Refer to 4.2.1, “Exporting performance extracts” on page 108, for a description of how to export performance extracts from which to create a chart. After exporting the extract, we used
file transfer to store the extracted data as a text file. We created directory `c:\reports\msexcel` to store all files that belong to Microsoft Excel chart project.

4.3.2 Importing extracted data into Microsoft Excel

To start Microsoft Excel, we clicked **Start->Microsoft Excel**. We wanted to import the text file that contains our delimited data to Microsoft Excel. To import the file, we clicked **File->Open**. The open window displays. We navigated to the directory that contains the file that we are going to open. The file is named `excel_chart.txt`.

We clicked **Open**. The Text Import Wizard window (Figure 4-8) opens.

![Figure 4-8 Text Import Wizard (Part 1 of 3)](image)

Since our extracted performance data is delimited by semicolon, we chose **Delimited** by selecting the radio button as shown in Figure 4-8. We wanted to start from row 1, which is specified by default. We clicked **Next** to continue.

The next Text Import Wizard window (Figure 4-9) opens. We selected the check box **Semicolon** and deselected the rest of the check boxes. When we selected check box Semicolon, the title line was aligned automatically by Microsoft Excel.
We checked that the data of the imported columns appeared as expected. After that, we clicked **Next**. Then the third Text Import Wizard window (not shown) opens. It allows you to select each column and set the data format. The default is fine. Therefore, we clicked **Finish** to continue and create our chart. Microsoft Excel displays the excel_chart data sheet as shown in Figure 4-10.

As shown in Figure 4-10, column B displays the average response time per transaction while column C displays the maximum response time per transaction. To create a visual comparison of average response time to maximum response time per transaction, we selected the area that is to be used as input data for the chart.
When we selected the area, we clicked the chart wizard icon as shown in Figure 4-11.

Microsoft Excel provides a chart wizard that guides you to create charts in four steps:
1. On the first window, you select the chart type. Each chart type has a number of sub types that you can select. You can also immediately see how your chart will look by clicking a sample button.
2. Specify the columns and labels you want to appear in your chart. If you already selected your columns and labels before you start the chart wizard, the preview of your chart may be correct so that you may not need to change anything.
3. Set standard option for your chart. You can check the effect of setting the options by looking at a preview chart.
4. Decide whether to place the newly created chart in a new sheet or as an object in a data sheet.

We clicked the chart wizard icon in the tool bar.

![Figure 4-11](predicted.png)

When the Chart Wizard window (Figure 4-12) opened, we selected a chart sub-type of clustered column with 3-D visual effect. You can select different chart types as well. We clicked Press and Hold to View Sample to check if the sub-type that we chose looks as expected. We clicked Next to continue.
When we started to work with the chart wizard, we already selected the data range of average response time and maximum response time. Therefore, the preview chart shown in Figure 4-13 looks correct. The preview chart shows a comparison of the selected cells. If the preview chart did not look as expected, you can click the Data Range tab to select cells that contain the input data required.

We clicked the Series tab and modified the names of our series cells. To change the Series name, we clicked in the name box and cleared its contents. After that, we typed a new name, resp ave, for series one. The name changes when you click the item that you are about to change in Series pane. We repeated these steps to rename series 2 to resp max. We clicked Next to continue.
Figure 4-13  Chart Wizard (step 2 of 4)

The Chart Wizard window, step 3 of 4, opens. Step 3 of the chart wizard allows the setting of standard options for the chart that you created. For simplicity, we did not change anything during the process of step 3. The chart is therefore created as shown in the preview section of the Chart Wizard window in Figure 4-14.

Figure 4-14  Chart wizard (step 3 of 4)
We clicked **Next** to continue. Chart Wizard window (step 4 of 4) opens. During the process of step 4, we had to decide whether we wanted to place the chart as a new sheet or as an object on an existing data sheet. We kept the default and placed the chart as an object in the `excel_chart` data sheet. We clicked **Finish** to finally create our chart.

Figure 4-15 shows the chart that we created. The appearance of the chart can still be improved. You can change the way the data is shown, color, chart options, and chart type.

![Figure 4-15  Microsoft Excel chart](image)

**4.4 Processing extracts with Microsoft Access**

Microsoft Access provides a chart wizard that you can use to create charts based on the data specified in a form or report. CICS Performance Analyzer allows you to export performance extracts that can be imported to Microsoft Access. To use the chart wizard, you must have Microsoft Graph 2000 installed. The following sections describe step-by-step how to:

- Export performance extract
- Import extracted data to Microsoft Access
- Create a form based on the extracted data
- Use the chart wizard to create a chart in a form

**4.4.1 Exporting performance extract**

Refer to 4.2.1, “Exporting performance extracts” on page 108, for a description about how to export performance extracts. After exporting the extract, we used file transfer to store the performance extract as a text file. We created directory `c:\reports\msaccess` for that.

**4.4.2 Importing extracted data into Microsoft Access**

We clicked **Start->Programs->Microsoft Access**. When the main Create a New Database window opened, we clicked **Cancel**. To open our text file, we clicked **File->Open** and navigated to the directory that contains the `c:\reports\msaccess\acces_chart` file. We clicked **Open**.
Figure 4-16 shows the Link Text Wizard window that opened. We made sure that radio button **Delimited** was selected since our test file is delimited by semicolons. We clicked **Next**, which opened another page of the link text wizard.

![Link Text Wizard](image)

On the second page of the Link Text Wizard (Figure 4-17), we clicked the **Semicolon** radio button to specify which type of delimiter we used. It is important to select that the first row of our table contains field names. Otherwise Microsoft Access inserts a title line with generated field names. Therefore, we selected the **First Row Contains Field Names** option to avoid generating a second line of field names. We clicked **Next**.

![Link text wizard 2](image)
The next Link Text Wizard window allows the specification of information about the fields you are importing. You can associate field types, eliminate columns of the table, or do both. We kept the defaults and clicked **Next**.

On the last Link Text Wizard window we specified a name for the linked table and clicked **Finish**.

The Database window opened. On the left Object pane, we clicked **Forms** and then double-clicked **Create form by using wizard**.

The Form Wizard window (Figure 4-18) opened. We moved the fields we wanted in the form to the selected fields pane on the right in the window. All the fields can be moved, or only the ones that you want to use for the chart. We clicked **Next**.

![Form Wizard](image)

*Figure 4-18  Form Wizard*

The second page of the Form Wizard appears. It allows the selection of a layout for the form. We selected the **Data sheet** radio button and clicked **Next**.

The third page of the Form wizard displays. We kept the standard style and clicked **Next**.

On the next Form Wizard window, we typed the name of the form and clicked **Finish**.

The Database window opens again (Figure 4-19). We double-clicked the form name that we created. Our form name was **Access_chart**.
A data sheet window of form Access_chart opens. We can only use the chart wizard if the form will be displayed in design view mode. Therefore, while the form was still open, we clicked the **Design View icon** in the tool bar. The view of the form changed to design view.

Figure 4-20 shows the Access_chart form in design view mode. We clicked **Insert->Chart** while the form was displayed in design view.
The Chart Wizard window opened. We needed to specify which table or query we want to use to create the chart. `Access_chart`, the name of our table, is already selected. The radio button `Tables` is also selected. We made no further changes and clicked `Next`.

The second Chart Wizard window opened. Here, we selected which fields of the table contained the data for the chart. We moved the fields from which we wanted to create the chart to the right pane. We selected `Tran`, `response time`, and `suspend time`, which should allow a visual comparison of response time and suspend time per transaction. We selected the fields in the left pane and moved them to the right pane by clicking the button with the right arrow. When we finished moving the required fields to right pane, we clicked `Next`.

The third Chart Wizard window (Figure 4-21) opened. We selected a 3-D column chart. This allowed the comparison of data points along two axes and the comparison between items, `dispatch time`, and `response time`. When we chose the type of chart we wanted, we clicked `Next`. 

![Figure 4-20 Chart wizard invoked while in design view mode](image-url)
The next chart wizard window was displayed. We used drag and drop to move field buttons, response time, and dispatch time, from the right to the data pane on the left of the chart. When both field buttons were moved, we double-clicked the **Response time field** button and selected **Average** from the list of options. Then we double-clicked the **Dispatch time field** button in the data pane and selected **Average**.

Next, we used drag and drop to move the **Tran** field button from the right to the Axis pane on the left of the chart wizard window. No further customization needed to be done to the **Tran** field button. We clicked the **Preview Chart** button in the upper left corner of the Chart Wizard window to check if the basic layout of the chart is displayed correctly. Then we clicked **Next** to continue creating our chart.

The next Chart Wizard window (Figure 4-22) opens. We did not want the chart to change from record to record, so we clicked **Next**.
The next Chart Wizard window that opened allowed the changing of the name of the chart title. We kept the name that was displayed, which is Access_chart. We clicked Finish to complete the Chart Wizard process.

We returned to the Form Design View window, which displayed the chart. The chart indeed shows sample data rather than what we expected. We discovered that if you are in a form design view when you first create a chart, you have to switch to form view to see current data. Therefore, we clicked View->Form View which displays the Access_chart form view window including the chart that we created.

Figure 4-23 shows the form view of our chart. The chart is displayed based on a single record. You can move from record to record and you see a chart that represents only the data in the current record. Based on the data you specify, the chart wizard determines whether it should display data from all fields in one global chart, or whether it is more appropriate to show a record-bound chart. We actually wanted a global chart. Therefore we switched the form view back to design view by clicking View->Design View.

![Form view of a chart](image)

The chart was displayed in design view mode (Figure 4-24). We selected the chart and clicked the Properties icon in the tool bar. You can also right-click the selected chart and then choose Properties.

When we clicked the Properties icon, the Properties window opened. We clicked data (only if it is not already open) and located the property boxes, LinkChildFields and LinkMasterFields. Both fields are linked to the Tran field of our table.

To create global chart, we unlinked the Tran field from the chart. To unlink the field, we clicked the Properties box for LinkChildFields. When a small pop-up box opened, we clicked it as well, which cleared the Properties box.
After that we unlinked the LinkMasterfields properties box in the same way. We closed the Properties window and verified that the chart layout changed from a record-bound chart to a global chart.

Figure 4-24   Design view of Access_chart

We changed the view back to form view by clicking View->Form View. You can customize the appearance of the chart by double-clicking the chart. The border of the chart changes to a bold light gray color. We then clicked Chart in the tool bar to see a list of options to choose from. After that, we saved the form containing the chart.
This part goes through a sequence of scenarios that show how CICS Performance Analyzer (PA) reports and extracts can help you to analyze the behavior of your system or performance characteristics of specific applications. The scenarios we discuss cover traditional application workloads (Cobol, VSAM, DB2, MQ) as well as new Web-initiated or Java workloads.

This part also explains how you can use:
- CICS PA to evaluate the overall system performance
- A series of CICS PA reports to drill down on particular problems
- The Historical Database to maintain, report and export to DB2 CICS Monitoring Facility (CMF) performance data
System setup and scenario overview

This chapter describes the ITSO CICSplex environment that we used to run our sample scenarios and generate performance data. To generate transaction workloads, we combined several applications that were used in the course of several residency projects and described in other Redbooks. Consequently, our CICSplex configuration in this project is a combination of configurations used in other ITSO projects.

This chapter also introduces a set of scenarios that were used to demonstrate how you can use CICS Performance Analyzer (PA) to improve application performance, to verify application design, or to improve overall performance of a complex CICS system.
5.1 CICS region setup

We decided to build a configuration that would support a mixture of traditional (COBOL, VSAM, DB2) and new (Java, Enterprise JavaBean (EJB)) workloads. To simplify the CICSPlex System Manager (SM) setup, we built our CICSPlex using only one z/OS image, SC66, of the ITSO Parallel Sysplex®.

By convention, the APPLIDs of CICS regions are composed of an SCSC string concatenated with a SYSID. For example, our CICSPlex SM address space (CMAS) region has a SYSID of CMAS and the APPLID of SCSCCMAS. All CICS regions described in this chapter are connected and defined to our CMAS and run under its control.

5.1.1 Configuration for traditional workloads

To run traditional 3270 workloads, we defined two CICS Terminal Owning Regions (TORs) with SYSIDs of PTA1 and PTA2. The workload was generated using the Teleprocessing Network Simulator (TPNS) running on another z/OS image of our Parallel Sysplex, SC47. Both TORs registered with the VTAM Generic Resource facility, so the 3270 workload was balanced between the two TORs. All transactions can also be run manually from 3270 emulation screens. The CICS level of the TOR regions was CICS Transaction Server 1.3.

Each TOR had MRO connectivity to four Application Owning Regions (AORs). AOR regions PAA1 and PAA4 were at CICS Transaction Server (TS) 2.2 level and were used to run VSAM workload. The VSAM data sets used by the workload can be opened in Record Level Sharing (RLS) mode to be accessible with update integrity from both AORs at the same time. They can also be opened in Local Shared Resource (LSR) mode. However, if one of the two AORs opened them for update, another can only open them for read and browse operations. Dynamic transaction routing under control of CICSPlex SM was used to forward the VSAM workload to AORs.

In the problem determination scenario described in Chapter 18, “Using CICS Performance Analyzer reports for problem determination” on page 385, we had TPNS create a workload as though users were signing on to PAA1 directly, and PTA1 was used as an FOR to function ship file requests, and tuned for file IO appropriately.

AOR regions PJA6 and PJA7 were at CICS TS 2.2 level, and they were used to run DB2, Java, and EJB workloads. The DB2 and Java simulated workloads were also generated by TPNS using the same 3270 network definitions as those used for the VSAM workload. We used static transaction routing definitions to forward transactions that access DB2 tables from PTA1 to PJA7 and from PTA2 to PJA6 (see Chapter 7, “Tuning the CICS DB2 attachment facility” on page 171). The Java transaction was routed in a similar way (see Chapter 12, “Java applications in CICS” on page 267).

5.1.2 Configuration for Enterprise JavaBean workloads

To run EJB workloads, we built a logical EJB server composed of two Internet Inter-ORB Protocol (IIOP) listener regions, PLA1 and PLA2, and two Java AORs, PJA6 and PJA7. The workload was generated using a set of batch script files running on a workstation. All applications can also be run manually from a browser through a menu provided by a servlet that could be invoked in a WebSphere Application Server on a Windows 2000 workstation. In both cases, the client (command line client or servlet) invokes an Enterprise JavaBean running in CICS Java AORs. The CICS level of the IIOP listener regions and Java AORs was CICS TS 2.2.
To enable CICS to publish references to the home interfaces of enterprise beans and to enable enterprise bean clients to obtain these references using the Java Naming and Directory Interface (JNDI) API, we set up a name server using a z/OS Lightweight Directory Access Protocol (LDAP) server. This LDAP server can run on any MVS image, but we used it to start on the same image where our CICS regions ran.

Each of the listener regions had MRO connectivity to each of the Java AORs. There is also MRO connectivity between PJA6 and PJA7 because one of our enterprise bean applications required access to a partner AOR.

The listener regions used TCP/IP port sharing, so the IIOP inbound requests could be balanced between the listener regions. The distributed routing mechanism, implemented by CICSPlex SM provided EYU9XLOP program, was used to balance method requests across AORs PJA6 and PJA7.

The Java AORs are connected to a DB2 subsystem D7Q2 and they also share a set of VSAM data sets opened in RLS mode used by our enterprise bean application. There are two versions of an application that use VSAM RLS files and two versions using DB2 data (JDBC and SQLJ).

For a complete description of how to set up a CICS logical EJB server, refer to CICS Transaction Server for z/OS Java Applications in CICS, SC34-6000, and Chapter 5 in the IBM Redbooks publication Enterprise JavaBeans for z/OS and OS/390 CICS Transaction Server V2.2, SG24-6284. You can also refer to Chapters 10, 11, and 12 in the SG24-6284 book for a description of the ITSO enterprise bean Trader application.

### 5.1.3 CICS Web Support and 3270 Bridge setup

To demonstrate use of CICS PA in conjunction with CICS Web Support and 3270 Bridge, we used a stand-alone CICS TS V1.3 region PAA6 and the CICS TS V2.2 region PJA6. Applications were run manually from a browser window. Refer to Chapter 11, “CICS Web Support and 3270 Bridge” on page 257, for more details.

### 5.1.4 CICS Transaction Gateway setup

We also included two CICS Transaction Gateways in our configuration: one running on z/OS and another running on a Windows 2000 workstation. With the z/OS version of the gateway, an application can only use External CICS Interface (EXCI) to access CICS programs. With the distributed (Windows, for example) version of the gateway, an application can use External Call Interface (ECI) to access CICS programs or External Presentation Interface (EPI) to access CICS 3270 transactions. One of our applications running in WebSphere Application Server used ECI calls (not EPI) to invoke a CICS program. When the z/OS gateway is used, the ECI calls are mapped to EXCI calls that provide the same functionality, so our application could access CICS through either one of the gateways.

Both varieties of the CICS Transaction Gateway allow limited workload management between CICS regions to which the gateway is connected. We decided to define one upstream connection for each of the gateways. We connected the z/OS gateway to region PJA6 and the distributed gateway to region PJA7. The distributed gateway was connected using the new ECI over TCP/IP communication mechanism introduced in CICS TS V2.2.

As was the case with the IIOP workload, the workload through gateways was generated using a set of batch script files running on a workstation. The application can also be run manually from a browser through a menu provided by a servlet that can be invoked in a WebSphere Application Server on a Windows 2000 workstation. In both cases, an Enterprise
JavaBean running in WebSphere Application Server on Windows 2000 invokes a CICS program through the gateway.

For a description of how to set up the CICS Transaction Gateway on various platforms, refer to *CICS Transaction Gateway V5: The WebSphere Connector for CICS*, SG24-6133. For a description of the ECIRequest version of the ITCO Trader application, refer to Chapter 10 of *Enterprise JavaBeans for z/OS and OS/390 CICS Transaction Server V2.2*, SG24-6284.

Figure 5-1 shows our connectivity setup. MRO connections from all the CICS regions to the CMAS, as well as IP connections between Java AORs (PJA6 and PJA7) and the LDAP server, are not shown on the chart.

*Figure 5-1  CICS region setup*

Figure 5-2 shows how our AORs accessed VSAM and DB2 data.
Table 5-1 summarizes the functions of different CICS regions in our setup.

<table>
<thead>
<tr>
<th>CICS APPLID</th>
<th>CICS SYSID</th>
<th>Function and CICS TS level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCSCCMAS</td>
<td>CMAS</td>
<td>CICSPlex SM CMAS (2.2)</td>
</tr>
<tr>
<td>SCSCPTA1</td>
<td>PTA1</td>
<td>Terminal Owning Region (1.3)</td>
</tr>
<tr>
<td>SCSCPTA2</td>
<td>PTA2</td>
<td>Terminal Owning Region (1.3)</td>
</tr>
<tr>
<td>SCSCPLA1</td>
<td>PLA1</td>
<td>IIOP Listener Region (2.2)</td>
</tr>
<tr>
<td>SCSCPLA2</td>
<td>PLA2</td>
<td>IIOP Listener Region (2.2)</td>
</tr>
<tr>
<td>SCSCPAA1</td>
<td>PAA1</td>
<td>Application Owning Region (1.3)</td>
</tr>
<tr>
<td>SCSCPAA4</td>
<td>PAA4</td>
<td>Application Owning Region (1.3)</td>
</tr>
<tr>
<td>SCSCPAA6</td>
<td>PAA6</td>
<td>Application Owning Region (1.3)</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>PJA6</td>
<td>Application Owning Region (2.2)</td>
</tr>
<tr>
<td>SCSCPJA7</td>
<td>PJA7</td>
<td>Application Owning Region (2.2)</td>
</tr>
</tbody>
</table>

5.2 Scenarios

This section provides a brief description of scenarios that we used to demonstrate how you can use CICS PA to improve performance of a given application or analyze performance of a complex CICS system.

**Note:** The scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer reports and extracts. The CICS regions were not necessarily tuned for peak performance. And in some cases, a high level of tracing was active. Therefore, these scenarios and the results are provided for demonstration purposes only. They do not provide definitive results for a customer environment.
5.2.1 VSAM application performance analysis and Transaction Resource Monitoring support

In this scenario (refer to Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139), we generate the following CICS PA reports:

- Performance Summary report
- Performance List report
- Performance List Extended report
- Transaction Resource Usage report
- Transaction File Usage Summary report

We use them to compare execution characteristics of an application that is composed of a set of nine transactions that access nine VSAM data sets. You can access the VSAM data sets in either Local Shared Resources (LSR) or Record Level Sharing (RLS) mode.

This approach can be useful if you want to determine whether LSR or RLS access will provide better response time in a case of a specific application in your environment. We also show how the CICS PA reports can help you tune your LSR pool.

5.2.2 Tuning the CICS-DB2 attachment facility

In this scenario (refer to Chapter 7, “Tuning the CICS DB2 attachment facility” on page 171), we generate the following CICS PA reports:

- DB2 Short Summary report
- DB2 Long Summary report
- DB2 Recap report
- DB2 List report

We use them to adjust values of parameters that influence performance of the CICS DB2 attachment facility. We also show the difference in usage of system resources between a threadsafe and a non-threadsafe environment.

5.2.3 WebSphere MQ

In this scenario (refer to Chapter 8, “WebSphere MQ” on page 205), we focused on the following WebSphere MQ reports:

- Class 1 List and Summary reports
- Class 3 List and Summary reports

We use these report to determine which kind of the MQ API requests are issued in CICS applications. We relate the WebSphere MQ reports to a CICS performance report and show the Transaction Temporary Storage Usage Summary report.

5.2.4 CICS use of MVS System Logger

In this scenario (refer to Chapter 9, “CICS and MVS System Logger” on page 223), we generate the following CICS PA reports:

- System Logger Logstream Summary report
- System Logger List report

We use these reports to identify possible areas of improvement with regards to allocation of a direct access storage device (DASD)-only log stream for use by the CICS logger component.
5.2.5 CICS access through CICS Transaction Gateway

In this scenario (refer to Chapter 10, “Scenarios with CICS Transaction Gateway” on page 241), we generate the following CICS PA reports:

- Performance Summary report
- Performance List report

We use them to compare internal CICS response times and consumption of system resources by a CICS application that is invoked from the WebSphere Application Server environment in two cases:

- CICS Transaction Gateway for z/OS
- CICS Transaction Gateway for Windows 2000 accessing CICS through ECI over TCP/IP

We also show how CICS PA can help us to determine how a CICS client application uses the ECI interface and suggest a possible improvement.

5.2.6 CICS Web Support and 3270 Bridge

In this scenario (refer to Chapter 11, “CICS Web Support and 3270 Bridge” on page 257), we generate the following CICS PA reports:

- Transaction Group report
- Transaction Group Summary report

We use these reports to tune our CWS environment for better performance.

5.2.7 Java applications in CICS

In this scenario (refer to Chapter 12, “Java applications in CICS” on page 267), we generate the following CICS PA reports:

- Performance Summary report
- Performance List report

We use these reports to:

- Compare behavior of a CICS Java application in the environment of a resettable Java Virtual Machine (JVM) to the behavior in the environment of a non-resettable JVM.
- Show the advantage of using shared application classpaths.

5.2.8 Enterprise JavaBeans in CICS

In this scenario (refer to Chapter 13, “Enterprise JavaBeans in CICS” on page 285), we generate the following CICS PA reports:

- Performance List report
- Performance Summary report

We use these reports to determine the optimum number of JVMs to be run in our CICS regions. We also use these reports to compare consumption of system resources by the same CICS enterprise bean application in three different data access cases:

- Access to VSAM data sets through JCICS classes
- Access to DB2 data through JDBC interface
- Access to DB2 data through SQLJ interface
5.2.9 Application Naming support

In this scenario (refer to Chapter 14, “Application Naming support” on page 311), we show how to implement the new Application Naming support introduced by CICS TS V2.2 authorized program analysis report (APAR) PQ63143 and CICS TS V1.3 APAR PQ63141. We use the CICS PA Performance List report.

5.2.10 CALL and LINK performance

In this scenario (refer to Chapter 15, “CALL and LINK performance” on page 319), we compare CPU consumption of an EXEC CICS LINK command to that of a CALL command when used in a COBOL language program in LE/370 environment. We also show how to use event monitoring points to add user fields to CMF performance records and how to handle these fields with CICS PA. We use the CICS PA Performance List report.

5.2.11 Exception reporting

In this scenario (refer to Chapter 16, “Exception reporting” on page 343), we show how to generate CICS PA exception reports:

- Exception List report
- Exception Summary report

5.2.12 Analyzing overall CICS system performance

In this scenario (refer to Chapter 17, “Analyzing overall system performance” on page 355), we use various CICS PA reports to analyze overall performance of a complex CICS system that runs a mix of transactions that have different execution characteristics. We generate the following CICS PA reports:

- Performance List report
- Cross-System Work report
- Workload Activity report

5.2.13 Using CICS Performance Analyzer reports for problem determination

In this scenario (refer to Chapter 18, “Using CICS Performance Analyzer reports for problem determination” on page 385), we use various CICS PA reports to analyze the impact that an application change makes to our CICS system. We identify problems introduced by the change, and through a series of reports, can identify the program changes responsible for the problem. We generate the following CICS PA reports:

- Performance Summary report
- Wait Analysis report
- Cross-System report
- Transaction Resource Usage List report
- Performance List report
5.2.14 Historical Database (HDB)

In this scenario (refer to Chapter 19, “Historical Database” on page 415), we show how to use all options from the Historical Database. A Summary HDB is used to demonstrate:

- Template
- Define
- Load
- Report
- Maintenance
- Housekeeping

A List HDB is used to show the new export function for loading CICS performance data from SMF records into a DB2 table.
Chapter 6. VSAM application performance analysis and Transaction Resource Monitoring support

This chapter demonstrates how we used the capabilities of CICS Performance Analyzer (PA) to measure the achieved performance objectives of a 3270 VSAM RLS/LSR application scenario. The various CICS Performance Analyzer tasks that are necessary to measure the performance data are described in detail.

It also discusses and uses the new Transaction Resource Monitoring feature introduced by CICS Transaction Server (TS) V2.2 authorized program analysis report (APAR) PQ63143. The equivalent CICS TS V1.3 APAR is PQ63141. For a description of Application Naming support, refer to Chapter 14, “Application Naming support” on page 311.

When the Transaction Resource Monitoring feature was introduced by APARs PQ63143 and PQ63141, the default for the DFHMCT TYPE=INITIAL FILE parameter was 0. The default value is now 8. This new default is introduced by CICS TS V2.2 APAR PQ76701. The equivalent CICS TS V1.3 APAR is PQ76695.

Transaction Resource monitoring is enhanced further with APARs PQ76703 and PQ76698 for CICS TS V2.2 and CICS TS V1.3 respectively. These APARs permit writing monitoring resource class records for resource managers such as DB2 and DBCTL used by transactions.

Refer to Table 2-1 on page 26 for a concise list of the enabling program temporary fixes (PTFs).

**Note:** These scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer reports when running with various VSAM workloads. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, these scenarios and the results provided for demonstration only. They do not provide definitive results for a customer environment.
6.1 CICS VSAM interface

CICS file control supports three VSAM access modes:

- **Local shared resources (LSR):** These files share a common pool of buffers as well as a common pool of strings.
- **Nonshared resources (NSR):** CICS files that are defined as nonshared have their own buffers and strings.
- **VSAM Record Level Sharing (RLS):** Unlike LSR or NSR access modes, VSAM RLS allows multiple CICS TS regions to access VSAM files for update with full data integrity.

In a traditional CICSPlex system setup consisting of Terminal Owning Regions (TORs), Application Owning Regions (AORs), and File Owning Regions (FORs), CICS LSR files and NSR files can be shared by using function shipping of file control requests to the FOR region. Function shipping to FOR regions can be processed with data integrity.

Using a single FOR in your CICSPlex had some disadvantages. When you use an FOR, there is a single point of failure. If the FOR region is lost for whatever reason, you have no access to your VSAM data sets. FORs cannot help if you need to share data sets between CICS and batch regions.

RLS support was introduced by DFSMS 1.3, and CICS supports RLS access from CICS TS V1.1. VSAM RLS allows one to share VSAM files sysplex-wide among multiple CICS TS AOR regions with full update integrity. Nonrecoverable files can be read and updated by CICS and batch jobs concurrently.

We used the CICS Performance Analyzer to investigate the performance behavior of a set of CICS transactions that access VSAM data sets in LSR or RLS mode.

---

**Important:** A new function, Transaction Resource Monitoring, allows collection of information about individual files, and can give a breakdown of file usage by transaction ID. For more information about this new function and Application Naming support, refer to Chapter 14, “Application Naming support” on page 311, which describes these new functions in greater detail.

---

6.2 CICS VSAM RLS scenario description

To produce useful and realistic performance data for the VSAM performance scenario project, we used a traditional 3270 COBOL VSAM application that was used previously in other Redbooks projects. The application is using nine VSAM files which can be opened in either RLS or LSR mode. Therefore, we investigated both an RLS and an LSR VSAM performance scenario. The 3270 VSAM application can be invoked using another complete set of definitions and resources. Therefore, each function of the application exists twice. All the resource definition names of the alternative set of definitions have an X character in their names. When we use both sets of definitions to invoke the VSAM application, we use eighteen VSAM files for the entire application.

We used the Teleprocessing Network Simulator (TPNS) to simulate a realistic 3270 workload environment.

The 3270 VSAM application workload consists of four business applications:

- **Hotels:** This is a simple 3270 hotel reservation application using two VSAM data sets. There are four transactions available which drive the application: HR1, HR2, HX1, and HX2.
Chapter 6. VSAM application performance analysis and Transaction Resource Monitoring support

Inventor: This is an inventory tracking application. It is mainly using four transactions to manage the inventory. Transactions IT8/IX8 update the inventory, and transactions IT2/IX2 are used to inquire about part locations within the inventory.

Specification: This is a bill of material management application. It is using four transactions (PS2/PX2 and PS3/PX3) to inquire information about part lists.

Stock: This is a stock control application. Transactions SC2/SX2 are used to update any inventory, transactions SC4/SX4 allow you to insert new vendors, and transaction SC6/SX6 is used to delete parts from the stock.

Figure 6-1 illustrates the VSAM RLS/LSR file usage of the entire workload. We do not describe non-VSAM-related aspects of the workload, such as program and map design, since they are not relevant to our VSAM performance scenario.

Refer to Chapter 5, “System setup and scenario overview” on page 129, for a full description of the CICS environment setup we used to run the workload in RLS or LSR mode.

6.2.1 RLS workload generation

We used Teleprocessing Network Simulator (TPNS) to generate an appropriate 3270 VSAM workload. TPNS is a terminal and network simulation tool. Apart from a number of other functions, it can be used to simulate 3270 terminal operator input. This is done by arranging a TPNS script. A TPNS script consists of a message generation deck and a network definition. There is one message generation deck available for each transaction. It controls the messages from the simulated device to the subsystem. The network definition describes the terminal characteristics for each terminal that is to be simulated.

We arranged a message generation deck for each transaction as well as a network definition for two hundred 3270 terminals. This allows us to simulate two hundred 3270 terminals that
can invoke transactions that compose our VSAM application. To run some transactions more often than others, we used the PATH statement to define a sequence of transactions for each terminal to process. For each transaction in the path, we specified a number in a DIST statement that corresponds to the entry in the path.

Example 6-1 shows how we specified the probability of distribution for our workload scenario. TPNS adds all the numbers in the DIST statement. After that, it generates a random number between 1 and the sum. TPNS then chooses the deck that has the number in its range.

Example 6-1  Probability of distribution

<table>
<thead>
<tr>
<th></th>
<th>PATH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PS3,PS2,IT8,IT1,IT2,HR1, SC6,SC2,SC4,T51, PX3,PX2,IX8,IX1,IX2,HX1, SX6,SX2,SX4,TX1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIST</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>90,100,80,30,60,30,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100,30,20,40,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>90,100,80,30,60,30,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100,30,20,40</td>
<td></td>
</tr>
</tbody>
</table>

6.2.2 Performance objectives

The performance objectives and priorities depend very much on user expectations. From our previous experiences with the application, we know what average response times we can expect for different transactions. From the point of view of a terminal user, this is the most important characteristic. Another priority we have is that the application run without any resource constraints to avoid deadlocks, waits, or abends. We expect the workload we generate with TPNS to run smoothly and hope to achieve maximum throughput.

Figure 6-1 summarizes our performance objectives.

Table 6-1  VSAM scenario performance objectives

<table>
<thead>
<tr>
<th>Application</th>
<th>Transaction</th>
<th>Response ave</th>
<th>Response max</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOTEL</td>
<td>HR1 HX1</td>
<td>&lt; 0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>INVENTOR</td>
<td>IT8,IX8</td>
<td>&lt;0.15</td>
<td>0.2</td>
</tr>
<tr>
<td>INVENTOR</td>
<td>IT2,IX2</td>
<td>&lt;0.015</td>
<td>0.02</td>
</tr>
<tr>
<td>SPECIFIC</td>
<td>PS2,PS3,</td>
<td>&lt;0.015</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>PX2,PX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOCK</td>
<td>SC2,SC4,SC6, SX2,SX4,SX6</td>
<td>&lt;0.04</td>
<td>0.06</td>
</tr>
</tbody>
</table>

We are ready now to start the VSAM workload using TPNS. We gather the necessary performance data and analyze the results using a CICS Performance Analyzer summary report.
6.3 Running the VSAM RLS scenario

This section describes the tasks that we performed to collect the performance data for our VSAM application. First, we used CICS Performance Analyzer functionality to provide an overview of the performance behavior of the application. We performed the following steps:

1. Start TPNS and run the entire application for about 15 minutes. The application runs through TORs PTA1, PTA2 and AORs PAA1, PAA4.
2. Switch storage management subsystem (SMF) data sets. Note the name of the archived SMF data set that contains our performance data.
3. Create a summary performance report using CICS Performance Analyzer:
   a. Create a Summary Report Form
   b. Create an Object List
   c. Create a Report Set
4. Submit generated JCL and check the results.

6.3.1 Updating system definitions

After we ran TPNS for about 15 minutes, we switched SMF data sets and recorded the name of the archived SMF data set that contains the necessary performance records. The name of the data set is SMFDATA.ALLRECS.G8328V00.

CICS Performance Analyzer can automatically populate your systems definitions with details extracted from SMF files. To check that we are using the correct SMF data set and that we collected performance data for all the systems, we used the Take-up function first.

To perform the data Take-Up from SMF, we followed these steps:

1. From the CICS Performance Analyzer Primary Option Menu, we selected option 1.
2. Then the System Definitions menu opens. We selected option 4.
3. The Data Take-Up from SMF screen is displayed. We updated this screen with the SMF data set name as shown in Figure 6-2. Then, we pressed the Enter key to generate a batch job to extract the details from the SMF data set.

--- Figure 6-2 Data Take-Up from SMF screen ---
When the job completed, we were prompted to update your system definitions with the results of the batch job. The prompt opens when we reach the System Definitions screen the next time.

When the generated batch job has completed, we verified that we had SMF records for MVS image SC66 as well as for the CICS systems participating in the workload. We also checked that the time interval of our TPNS run was processed using this SMF data set. Figure 6-3 shows the output that was produced by the Take-up job.

Take-up processing updates the system definitions automatically. We invoked the Systems Definitions screen by selecting option 1 from the System Definitions Menu. Figure 6-4 shows the System Definition screen after processing Take-Up from SMF. It shows that the CICS systems we used were added automatically during take-up processing.

After that, we looked at the CICS System Definition entry for one of our AORs. We entered the S line command next to our AOR SCSCPAA1 listed on the System Definition screen.
The CICS System screen opens. We did not change any information on the CICS System screen. The name of the SMF data set was added to the screen automatically. For the moment, we did not want to process any user fields in our reporting. Therefore we did not need to specify further information like the name of an MCT or MCT load library. The default MCT is sufficient for now so all the information that was set by Take-Up from SMF processing is fine.

### 6.3.2 Creating a Summary Report Form

To see an overview of the performance behavior of our VSAM application, we had to arrange a Summary Report Form to view a performance summary of average response time, suspend time, and VSAM wait time. The format and content of the Summary Report can be customized using the Summary Report Form screen. Refer to 2.10.1, “Creating the Report Forms data set” on page 44, for a complete description of how to use Report Forms.

Figure 6-5 shows the fields that we moved above the end of report line. These are the fields that will fit in the report title line. They are included in the same order as they appear in the list. All the fields below the end of report line are ignored.

![Figure 6-5   VSAM application Performance Summary fields](image)

We created a new Report Form by selecting option 3 from the Primary Option Menu. In the command line, we typed `NEW` to create a new Report Form. The New Report Form screen opens, as shown in Figure 6-6. We entered a new name for the Report Form which is VSAMSUM. Then we typed the CICS APPLID. The CICS version was updated automatically.
Fields from CICS versions higher than 130 will not be available so use care as to which report is to be used when there are SMF records from mixed environments (refer to 17.2, “Working with different CICS system releases” on page 358, for further discussion).

We completed the new Report Form screen and pressed Enter. The Summary Report Form screen opens. A set of line commands can be used now to move fields that should appear in the summary report above the end of report line. Unwanted fields can be either deleted or moved below the end of report line. The end of extract line shown in Figure 6-5 is not used for Performance Summary reports.

6.3.3 Creating an Object List

Object Lists are introduced in 2.12, “Maintaining Object Lists” on page 56. This section describes how we created an Object List definition for the VSAM application. For the VSAM application scenario, we use 20 transaction definitions. Object Lists allow us to define a set of values that we can use to specify selection criteria for filtering SMF data for your reports. Therefore an Object List can be used to define all the transactions that belong to the VSAM application and use it as selection criteria in all reports that we will process to investigate the application performance.

To create an Object List for the VSAM application, we completed these steps:
1. From the Primary Option Menu, we selected option 4 (Object Lists).
2. The Object Lists screen opens. We entered the NEW command to create a new Object List. Figure 6-7 shows the new Object List screen we used to create the VSAMTRAN Object List. We pressed Enter to create the Object List and display the edit screen.
3. We added the first transaction name in the first Value field. Line action I can be used to insert another line for entry of the second transaction.

4. We continued until we added all transactions that belong to the VSAM application.

5. We pressed PF3 to save the new Object List.

### 6.3.4 Creating a Report Set

Finally, we needed to create a Report Set to run the performance summary process. From the Primary Option Menu, we typed option 2. The Reports Sets screen opens. In the command line, we typed NEW to create a new Report Set. The New Report Set screen opens. In the name field, we typed a new name for the Report Set which is VSAMSUM. Figure 6-8 shows the Edit Report Set screen.

![New Object List screen](image)

**Object Lists**

Command ===>

Specify the name of the new Object List and optional model.

Name . . . VSAMTRAN

Model ..

![Edit Report Set screen](image)

We type line action S next to the Summary Report field. Figure 6-9 shows the Performance Summary Report screen.
We entered the name of the CICS system in the APPLID field. Image SC66 was updated automatically. Under the heading Report Format, we pressed F4 to see the prompt of the available report formats. We created Report Form VSAMSUM earlier, so it should be in the list and can be selected. After that, we specified line action S in the Performance field under Selection Criteria.

Figure 6-9  Performance Summary Report screen

We pressed Enter in the Performance Summary Report screen. The Performance Select Statement screen is displayed. To use the Object List that we created earlier, we modified the bottom line of Figure 6-10. We specified the name of the Object List which was VSAMTRAN. VSAMTRAN contains objects of type TRAN. Therefore, we specified INC and TRAN, which means that we want to include performance records if or when the selection criteria is met.

Figure 6-10  Performance Select Statement screen

After that, we continuously pressed PF3 until we finally returned to the Edit Report Set screen. We are now ready to submit the job to produce the Performance Summary report.

When the job has completed, it produced the output shown in Figure 6-11.
When we looked at the output that was produced by the CICS Performance Analyzer Performance Summary report, we found that it pretty much complied with the expectations documented in 6.2.2, “Performance objectives” on page 142. There was one exception. The average response time of transaction IX8 was longer than expected. The average RLS wait time of 0.1139 seconds is more or less as expected but the average suspend time seems to be above our expectations. The average RLS wait time is a component of the average suspend time. We compared average RLS wait time with average suspend time of the other transactions that are part of the VSAM application. The average suspend time of transaction IX8 is worth a closer look.

Before we produced further reports to find the reasons for the increased suspend time, we wanted to produce a chart of the performance behavior first. If you have to present the results of your application performance tests, charts are visually appealing and make it easier for people to compare the performance behavior. It may be easier to look at charts to see whether performance improved rather than investigate many rows and columns of a large performance report.

We created a chart that displays average response time, average dispatch time, and average suspend time per transaction. We performed the following steps to create the chart that is shown in Figure 6-14.

1. Create a Report Set to export a performance extract.
2. Download the extract data set to a workstation as a text file.

The text file was imported to Microsoft Excel.

We created a chart based on the imported extract. From the CICS Performance Analyzer Primary Options Menu, we typed option 2 (Report Sets). In the command line, we typed NEW to create a new Report Set. We give our new report set a name, then press the Enter key. The Edit Report Set screen appears. Under heading Extracts, we typed line action S next to the Export option. The Export screen is shown in Figure 6-12.
We specified the CICS APPLID SCSCPAA1 because we used the region in our previous performance summary report. An output data set must be specified. We used EXTVSM1 and specified disposition 2. Under heading Extract Format, we pressed PF4 to see a list of available Report Forms. We used Report Form VSAMSUM that we created earlier. We kept the semicolon as a delimiter since it works well with Microsoft Excel. After that, we typed line action S next to the Performance option under Selection Criteria. We specified the same selection criteria that we used in Figure 6-10 to select all transactions that belong to the VSAM application. We then pressed PF3 until we returned to the Edit Report Set screen. After that, we typed RUN in the command line to generate the performance extract JCL. When the job completed, we had the extracted performance data available in the EXTVSM1 data set.

The data set can be downloaded to the workstation using the File Transfer utility of Personal Communications emulator or using FTP. We used the File Transfer utility to download the extracted data as a text file to the workstation.

Figure 6-13 shows the extracted performance data of our VSAM application which can be imported into any spreadsheet application in order to produce meaningful charts. Note that the header line is not aligned with the rest of the data. We found that the spreadsheet applications we used were able to align them automatically.
We used Microsoft Excel to create the chart shown in Figure 6-14 from the spreadsheet. It illustrates average dispatch time versus average suspend time per transaction. Average suspend time plus average dispatch time is the average response time as shown in the chart. The spreadsheet illustrates visually that there appears to be a problem with transaction IX8.

Next, we wanted to look more closely at the performance behavior of transaction IX8. To do that, we wanted to create a CICS Performance Analyzer Performance List report. The necessary steps to get such a report are very similar to ones we performed to create a Performance Summary report.

We wanted to keep the sequence of the performance fields that we specified in our summary Report Form VSAMSUM as shown in Figure 6-5 on page 145. We cannot use a summary form to format Performance List reports. Therefore we selected option 3 (Report Forms) from...
the Primary Option Menu. The Report Forms screen opens. We typed NEW in the command line to create a new Report Form. We entered a new name VSAML and selected option 1 to create a list type Report Form. After pressing the Enter key, the Edit List Report Form screen is displayed. A new sequence of performance fields can be arranged using the line commands to move the required fields above the end of report line. Refer to 6.3.2, “Creating a Summary Report Form” on page 145, to see the sequence of the VSAM related performance fields we used.

The performance list report for transaction IX8 can now be submitted. From the Primary Option Menu, we selected option 2 (Report Sets). The Report Sets screen shows a list of available Report Sets. We created Report Set VSAMSUM earlier, so it should be available in the list. We typed line action S next to the name of the VSAMSUM Report Set. The Edit Report Set screen appears.

We used Report Set VSAMSUM previously to create a Performance Summary report. Therefore, the summary item may still indicate yes as being active. We chose to deactivate the Performance Summary report before continuing. We did this by typing line action D next to the summary item.

To create a Performance List report, under the heading Performance Reports, we typed line action S next the List item. Figure 6-15 shows the Performance List Report screen.

```
File Systems Options Help

VSAMSUM - Performance List Report
Command ===>

System Selection: APPLID . . SCSCPAA1 + Report Output:
                      DDname . . . . . . . . . LIST0001
                      Print Lines per Page . . (1-255)

Image . +
Group . +

Report Format:
Form . . VSAML +
Title . .

Selection Criteria:
S Performance
```

Figure 6-15  Performance List Report screen
When we pressed Enter, the Performance Select Statement screen is displayed. It allows the specification of selection criteria for the performance list report. We selected transaction IX8 and a response time range of 2 - 10 seconds. This should result in a performance report that shows just transactions IX8 that have a response time longer than 2 and shorter than 10 seconds (Figure 6-16).

![Selection criteria for IX8 and the response time](image)

We returned to the Edit Report Set screen. We used the RUN line action command next to the List report to submit the job. Using the selection criteria shown in Figure 6-16, we produced the Performance List report shown in Figure 6-17. The Performance List report shows two IX8 tasks with bad response and suspend times. These tasks are likely responsible for the increased average response time that we discovered in our Performance Summary report.

![Performance list report using selection criteria](image)

So far, we knew that there was no general performance degradation. Just two transactions were suspended for a number of seconds. We wanted more detailed information about the reason for the suspension. We created a Performance List Extended report to collect further information about the situation.

The Performance List Extended report provides a detailed list of the CMF performance class records. It differs from the Performance List report in that you can specify the sorting criteria for the performance class records.

Before a Report Set can be submitted to request a Performance List Extended report, we had to create a new Report Form of the extended list type. From the Primary Option Menu, we selected option 3 (Report Forms). The Report Forms screen is displayed. We typed NEW in the command line to create a new Report Form for Extended List reports.
Report Form screen is displayed. We arranged the sequence of the performance fields that we wanted to fit in the title line of the report as illustrated in Figure 6-18.

![Figure 6-18 Sequence of performance fields for extended list reports](image)

After that we went back to the Primary Option Menu. From the Primary Option Menu, we selected option 2 (Report Sets). The Report Sets screen shows a list of the available Report Sets. We used line action S next to Report Set VSAMSUM to display the Edit Report Set screen. We deactivated any active reports by typing line action D next to them.

Figure 6-19 shows the Performance List Extended report screen we used to create an extended list report of our VSAM application. We updated the screen with information about the APPLID and the form name that we are going to use. Then we pressed Enter and the Performance Select Statement screen is displayed. We defined selection criteria for transaction ID IX8 and performance field TRAN.

For detailed information about how to specify a selection criteria on the Performance Select Statement screen, see 6.3.4, “Creating a Report Set” on page 147.
We pressed the PF3 key repeatedly to return to the Edit Report Set screen. We then entered the RUN line action command next to the List Extended report to submit the job.

Figure 6-20 shows the output of the Performance List extended process. Tasks #67075 and #67098 used additional resources because of the overhead of the abend processing for abend AEI9. It turned out that, while TPNS was running, two IX8 transactions were started manually. Since a MAPFAIL condition was not handled by the application, the tasks abended eventually.

We corrected the problem and started the VSAM RLS workload again using TPNS. The workload was running again for about 15 minutes when we switched SMF data sets and achieved a copy of the relevant SMF records. After that, we processed a Performance Summary report to check the average response time of all transactions that belong to the VSAM application workload.

We used the existing Report Set VSASUM to produce the Performance Summary report. We performed again the steps in 6.3.4, "Creating a Report Set" on page 147, to create the Performance Summary report using Object List VSOAMTRAN and Report Form VSUMSUM.
Object list VSAMTRAN was used as selection criteria to filter the transactions of the VSAM application. Report form VSMSUM was used to arrange the contents and the sequence of the title line on top of the Performance Summary report. Figure 6-21 shows that the average response time of transaction IX8 came down to 0.1183 seconds which is now in the range of our expectations.

![Figure 6-21 Performance Summary report results](image1)

To make it easier to see comparisons, we again extracted the performance data of the VSAM application and created a chart using Microsoft Excel (Figure 6-22).

![Figure 6-22 VSAM RLS application performance results](image2)
6.4 CICS VSAM LSR scenario description

When we finished analyzing the performance of the VSAM RLS application, we had to change the design of our environment slightly to perform a comparable performance analysis using LSR data sets (see Figure 6-23).

Since we do not use FORs anymore in our VSAM RLS environment, we eliminated AOR SCSCPAA4 from the LSR test environment. We did that because non-RLS files cannot be shared among CICS regions with update integrity. The environment we were going to use for the LSR performance scenario consists of TOR regions PTA1 and PTA2 and a single AOR PAA1. All VSAM application files will be allocated to AOR PAA1.

We logged on to AOR SCSCPAA1 and closed all VSAM application files that were still open in RLS access mode. To re-open them in non-RLS access mode, we entered the following command first:

```
CEMT SET DSN QUIESCE ALL
```

We installed a second set of the VSAM application file resource definitions which defined the files using the non-RLS access mode.

As we did for the RLS performance scenario, we used CICSPlex System Manager (SM) to manage our workload. The workload was generated using TPNS.

6.5 Running the VSAM LSR scenario

This section explains how we ran the LSR access mode scenario.

6.5.1 LSR workload generation

We used TPNS again to generate workload for the LSR performance scenario. The network deck had to be modified slightly to avoid generating transactions that belong to the first set of the VSAM application resource definitions as shown in Example 6-2. The second set of definitions can be used to invoke the application in RLS access mode as well as in LSR access mode. After modifying the TPNS network deck, a workload can be generated for transactions HX1, IX8, IX2, PX2, PX3, SX2, SX4, and SX8.
Example 6-2 Modified TPNS path and distribution probability

PATH PX3,PX2,IX8,IX1,IX2,HX1,
      SX6,SX2,SX4,TX1

DIST 90,100,80,30,60,30,
      100,30,20,40

TPNS starts the workload through TORs PTA1 and PTA2. Both TORs route the transactions
to AOR PAA1.

6.5.2 LSR application performance objectives

The VSAM LSR application that we plan to use for the CICS VSAM LSR performance
scenario was already used for a number of other projects. Therefore, we already had some
experience with its performance behavior. We never claimed that it was tuned to the
maximum performance, but we had some expectations about its response time and suspend
time. Since there is no service level agreement available for the application, we summarized
our expectations regarding the performance behavior, based on our previous experience, in
Table 6-2.

We expect the response times to be lower when the very same application is using LSR files
rather than RLS files. When we document the results of this scenario, we create a chart that
allows visual comparison between RLS and LSR response time.

Table 6-2 Performance expectations for an LSR scenario

<table>
<thead>
<tr>
<th>Application</th>
<th>Transaction</th>
<th>Response average</th>
<th>Response maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOTEL</td>
<td>HX1</td>
<td>&lt;0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>INVENTOR</td>
<td>IX8</td>
<td>&lt;0.15</td>
<td>0.2</td>
</tr>
<tr>
<td>INVENTOR</td>
<td>IX2</td>
<td>&lt;0.015</td>
<td>0.02</td>
</tr>
<tr>
<td>SPECIFIC</td>
<td>PX2,PX3</td>
<td>&lt;0.015</td>
<td>0.05</td>
</tr>
<tr>
<td>STOCK</td>
<td>SX2,SX4,SX6</td>
<td>&lt;0.04</td>
<td>0.06</td>
</tr>
</tbody>
</table>

6.5.3 Analyzing the current average response time

We started the TPNS workload and kept it running for about fifteen minutes. It takes about
three minutes for the TPNS CICS terminals to log on. While terminals are still in the process
of logging on to CICS, the first transactions of the workload started running. To find the
average response time of a smooth run, we used a time selection criteria to select
performance data after all terminals had completed logging on. Thus, we do not measure our
application performance data while additional CICS resources are consumed by the terminal
logon process.

After about 15 minutes, we stopped TPNS and issued an /I SMF command to switch the
active SMF data set. When the SMF data set was archived, we performed the following steps
to provide a CICS Performance Analyzer Performance Summary report:

1. Update system definitions.
2. Select Report Set VSAMSUM.
3. Deactivate any active reports.
4. Edit the performance summary report we used during the RLS scenario.
   a. Use existing Report Form VSAMSUM.
   b. Add a report interval to the performance select statement.
   c. Use Object List VSAMTRAN as a selection criteria to filter VSAM application transactions.

5. Run the Performance Summary report.

Updating system definitions

On the Primary Option Menu screen, we selected option 1 (System Definitions). The System Definition Menu opens. We selected option 4 (Take-Up from SMF file) and pressed Enter. On the next screen, we entered the name of the archived SMF data set and pressed Enter to submit the job.

```
--- System Definitions ---
Data Take-Up from SMF
Command ==>  

Data Set . . : 'SMFDATA.ALLRECS.G8446V00'
Instructions:
  Press ENTER to continue adding the systems.
  Enter DEFER command to defer adding the systems.
  Enter END or CANCEL command to cancel adding the systems.
```

*Figure 6-24  Take-Up from SMF update information*

When we return to the System Definition Menu, a screen opens indicating that system definitions were updated by the Take-Up from SMF function. Figure 6-24 shows the information screen of the take-up system update process. On the System Definition Menu, we selected option 1 (Define Systems, SMF Files and Groups) and pressed Enter. On the next screen, we typed line action S next to the system entry for AOR SCSCPAA1. We checked that the SMF data set was updated correctly. If it was not updated, we could insert it manually.

*Figure 6-25 shows the CICS system definition after running Take-Up from SMF.*
Selecting report VSAMSUM

We used Report Set VSAMSUM before to provide a Performance Summary report of our VSAM RLS application. On the Primary Option Menu, we select option 2 (Report Sets). The Report Sets screen displays a list of available Report Sets. We used line action S next to Report Set VSAMSUM. The Edit Report Set screen displays.

Deactivating any performance reports

Active performance reports have YES in the Active column on the Edit Report Set screen. We can enter line action D next to any active performance reports to deactivate them. Deactivating performance reports means that they will not be included in the current Report Set. They are still available for later use.

Editing the Performance Summary report

On the Edit Report Set screen, we typed line action S next to the Performance Report Summary option. We used summary reports before. Therefore the Summary Reports screen displays a list of summary reports that were created already. If no summary reports are available, the Summary Reports screen is bypassed and a screen to create the first summary report is displayed instead.

From the reports summary list, we selected the summary report that we created for the VSAM RLS scenario. The screen to edit the Performance Summary report is displayed. We checked that the correct APPLID is specified, which is SCSCPA1A1. Next, we specified the name of the Summary Report Form, which is VSAMSUM.

To specify a report interval and selection criteria, under Selection Criteria, we typed line action S next to the performance option. The screen displays a list of performance selection criteria or a screen that allows the editing of the performance select statement. We created selection criteria for Performance Summary reports before. A list of Performance Selection Criteria should be displayed. On the list, we typed line action S next to the selection criteria we used for the VSAM RLS scenario. Figure 6-26 shows the Performance Select Statement screen.
We included the start and stop time of the report interval that we were going to monitor. We started investigating SMF records when all terminals were logged on and the entire workload was running. After that, we included selection criteria specifying all the transactions that we used in the workload. We specified Object List VSAMTRAN that still contains all of the transactions of the workload. During the VSAM LSR performance scenario, we simply used the second set of transactions, which are the ones that have X characters in the middle of their name. We only had SMF records for the second set of transaction, so we can still use the Object List VSAMTRAN. There will not be any filter matches for the first set of transactions. We then pressed F3 repeatedly until we returned to the Edit Report Set screen.

Running the Report Set

On the Edit Report Set screen, we entered the RUN line action next to the Summary report and pressed Enter to display the Run Report Set screen. We did not change any information on the screen, and pressed Enter again to submit the job to generate the report.

6.5.4 Tuning changes to LSR

As you can see in Figure 6-27, response times are far higher than we expected. When we installed the CICS file resource definitions, we had to set up an LSRPOOL definition to share strings and buffers among the LSR files. We did not specify enough strings and buffers to run
the optimum number of tasks concurrently. Correct specification of number of VSAM strings and buffers is crucial for good performance of an LSR pool.

- **STRINGS** is used to determine the number of strings and thereby the number of concurrent operations possible against the LSR pool (assuming that there are buffers available).

- The number of buffers can have a significant effect on performance. The use of many buffers can permit multiple concurrent operations (if there are the corresponding number of VSAM strings available). It can also increase chances of successful buffer lookaside with the resulting reduction in physical I/O operations.

We specified the appropriate values for strings and buffers in our LSRPOOL definitions and invoked a performance summary report again.

### 6.5.5 Analyzing results

We applied the changes to the file resource definitions and started TPNS workload again. After about 15 minutes, we stopped TPNS and performed the following steps to generate a performance summary report again:

1. Update system definitions.
2. Select Report Set VSAMSUM.
4. Edit the Performance Summary report:
   a. Use the existing Report Form VSAMSUM.
   b. Update the report interval to the Performance select statement.
   c. Use Object List VSAMTRAN as selection criteria to filter VSAM application transactions.
5. Generate and submit performance list report JCL.

Figure 6-28 shows the Performance Summary report after we tuned the LSRPOOL. Response times and suspend times decreased significantly.

<table>
<thead>
<tr>
<th>Tran</th>
<th>Avg Response Time</th>
<th>Avg Dispatch Time</th>
<th>Avg User Time</th>
<th>Avg CPU Time</th>
<th>Avg Susp Time</th>
<th>Avg FC Total</th>
<th>Avg RLS Wait</th>
<th>Avg FC Wait</th>
<th>Avg FC ADD</th>
<th>Avg FCB ROW</th>
<th>Avg FC DELETE</th>
<th>Avg FC GET</th>
<th>Avg FC PUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HX1</td>
<td>.0126</td>
<td>.1730</td>
<td>.0028</td>
<td>.0017</td>
<td>.0098</td>
<td>2</td>
<td>.0000</td>
<td>.0066</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I1X</td>
<td>.0125</td>
<td>.2825</td>
<td>.0038</td>
<td>.0025</td>
<td>.0087</td>
<td>0</td>
<td>.0000</td>
<td>.0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I1X</td>
<td>.0315</td>
<td>.4220</td>
<td>.0042</td>
<td>.0038</td>
<td>.0272</td>
<td>17</td>
<td>.0000</td>
<td>.0202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I1X</td>
<td>.1477</td>
<td>.9045</td>
<td>.0052</td>
<td>.0032</td>
<td>.1425</td>
<td>14</td>
<td>.0000</td>
<td>.1297</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>PX2</td>
<td>.0160</td>
<td>.3446</td>
<td>.0043</td>
<td>.0031</td>
<td>.0117</td>
<td>18</td>
<td>.0000</td>
<td>.0067</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PX3</td>
<td>.0169</td>
<td>.4148</td>
<td>.0046</td>
<td>.0037</td>
<td>.0104</td>
<td>34</td>
<td>.0000</td>
<td>.0331</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S1X</td>
<td>.0437</td>
<td>.3044</td>
<td>.0046</td>
<td>.0037</td>
<td>.0391</td>
<td>18</td>
<td>.0000</td>
<td>.0331</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>S1X</td>
<td>.0128</td>
<td>.1965</td>
<td>.0048</td>
<td>.0027</td>
<td>.0000</td>
<td>9</td>
<td>.0000</td>
<td>.0046</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SX4</td>
<td>.0157</td>
<td>.2742</td>
<td>.0032</td>
<td>.0020</td>
<td>.0126</td>
<td>4</td>
<td>.0000</td>
<td>.0086</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>SX6</td>
<td>.0040</td>
<td>.1039</td>
<td>.0022</td>
<td>.0012</td>
<td>.0019</td>
<td>0</td>
<td>.0000</td>
<td>.0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 6-28 LSR Performance Summary report after tuning changes

To show visually the difference between the performance behavior before and after tuning the file resource definitions, we created a chart. The chart shown in Figure 6-29 compares response times per transaction before and after tuning the LSRPOOL.
6.6 Application performance RLS versus LSR

Finally, we used the Performance Summary reports for both of our scenarios (RLS and LSR) to extract data and create a chart that visually shows the comparison between response times in the case of RLS and LSR access mode. As we used the same application, data sets, and the same CICS environment for both scenarios, we can conclude that in our case the response times provided by RLS access are better than those provided by LSR access. The chart shown in Figure 6-30 illustrates the difference.

![Figure 6-30 VSAM RLS versus LSR](image-url)

6.7 Transaction Resource Monitoring

Transaction Resource is a new class of data for the CICS Monitoring Facility. The Transaction Resource data is collected at transaction termination for each resource specified in the MCT on the TYPE=INITIAL macro. A new system initialization parameter, MNRES=(OFF | ON),
specifies whether Transaction Resource Monitoring is to be made active during CICS initialization.

A new parameter has been added to the DFHMCT TYPE=INITIAL macro, FILE=(8 | number). This FILE option specifies the maximum number of files for which you want CICS to perform Transaction Resource Monitoring. This option is applicable only if Transaction Resource Monitoring is enabled, either by specifying MNRES=ON as a system initialization parameter (together with MN=ON), or by enabling it dynamically using either of the following commands:

EXEC CICS
CEMT SET MONITOR CICS

CICS standard monitoring performance class data includes totals for all files accessed by a transaction. Transaction Resource Monitoring collects information about individual files, up to the number of files specified. When the FILE parameter was first introduced, the default was FILE=0. With the code implemented by APARs PQ76701 and PQ76695, the default is now FILE=8.

The data collected is:
- File name
- Number and total time of file get requests
- Number and total time of file put requests
- Number and total time of file browse requests
- Number and total time of file add requests
- Number and total time of file delete requests
- Total number and total time of all requests against the file
- File access method request count
- File I/O wait time and number of waits
- RLS-mode file I/O wait time
- Coupling facility data table (CFDT) I/O wait time

The default value is 8 (eight).

Number specifies the maximum number of files, in the range 1 through 64, for which CICS is to perform Transaction Resource Monitoring. CICS collects monitoring performance data at the resource level for each file accessed by a transaction, up to the maximum specified by number. If the transaction accesses more files than the number specified, any files over the maximum are ignored, but a flag is set to indicate that the transaction has exceeded the file limit.

If you specify FILE=0, specifying MNRES=ON either as a system initialization parameter or dynamically while CICS is running, has no effect and Transaction Resource Monitoring data is not collected for files.

The monitoring domain exit, XMNOUT, is invoked at a new event point before a Transaction Resource Monitoring record is written to the transaction resource record buffer. This new invocation means that if performance class and Transaction Resource Monitoring are both active in your CICS region, XMNOUT can be invoked twice for the same event.

You can map the new transaction resource data using the new CICS monitoring domain copybook, DFHMNRDS.

The RESRCECLASS option (RESRCE | NORESRCE) was added to the CEMT INQUIRE and CEMT SET MONITOR commands to support the transaction resource class. The RESRCECLASS option has also been added to the EXEC CICS INQUIRE and SET MONITOR commands.
CICS PA and Transaction Resource Monitoring

You can use CICS PA to report on the new Transaction Resource Monitoring class records. We describe the process we went through to obtain a CICS PA Transaction Resource Usage Report.

We used TPNS to run our VSAM workload as discussed in Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139. After the workload completed, we switched the SMF data sets using the ISMF command. We took the copied SMF data set and added it to CICS PA. Figure 6-31 shows the data set added to the CICS regions SCSCPAA1 System Definition.

--- System Definitions ---

```
CICS System definition:
    APLID . . . . . . . . SCSCPAA1  MVS Image . . SC66
    Description . . . . . System added by take-up
    CICS Version (VRM) . . 620
    MCT Suffix . . . . . . I3
    MCT Load Library . . 'CICSSYSF.APPL62.LOADLIB'
    SDFHLOAD Library . . 'CICSTS22.CICS.SDFHLOAD'
    Dictionary DSN . . . . 'CICSL55.DFHMNREC'

/ Exc  SMF Data Set Name +    UNIT + SEQ VOLSER +
   'SMFDATA.ALLRECS.G8885V00'    DASD
```

--- End of list ---

Figure 6-31  SCSCPAA1 System Definition

We also included the MCT Suffix (I3), which was used at CICS Initialization. Example 6-3 shows the DFHMCT TYPE=INITIAL statement which includes the new option FILE.

```
Example 6-3  DFHMCT

I3  DFHMCT TYPE=INITIAL,
    APPLNAME=YES,
    FILE=9,
    SUFFIX=I3 *
    *
    *
```

We then needed to create a new Report Set to report on the transaction resource data. We did this by specifying NEW TRANSREP on the Report Sets screen. When the new Report Set was displayed, we selected the File Usage Summary Report option (Figure 6-32).
On the next screen, the File Usage Summary Report, we selected the default options, that is, Summary Reports for Transaction File Usage, File Usage broken down by transaction IDs and transaction totals (Figure 6-33).

**Figure 6-32  Select Transaction Resource Usage: File Usage Summary**

**Figure 6-33  File Usage Summary Report options**
We pressed the PF3 key until we returned to the Edit Report Set screen, used the RUN line action command on the Transaction Resource Usage List report, added the CICS APPLID to the Run Report Set screen, and submitted the job. Figure 6-34 shows part of the report produced from this run. It shows two transaction (HX2 and IT2). HX2 made file requests to file HOTEL1X and IT2 made file requests to something called *CTLACB* and ITEMACT. The *CTLACB* indicates that we suffered a file control wait. The resource name of *CTLACB* shows that the resource being waited for is the RLS Control ACB.

**Figure 6-34  Transaction File Usage Summary report**

Figure 6-35 shows the File Usage Summary report, a summary of File usage broken down by transaction ID.

**Figure 6-35  File Usage Summary report**
In this report, we the file HOTEL1X is displayed with the two transactions, HX1 and HX2, which made a sort of file request to the file. Following is the total of all transactions file requests for the file being reported.

We then ran the Transaction Resource Usage List report by selecting it from the Report Set screen. When we did this, the Transaction Resource Usage Report screen (Figure 6-36) is displayed. We selected the File Usage report in the Detailed List Reports Required section, pressed F3 until we returned to the Reports Set screen, and then used the RUN line action on the report to submit the job.

![Figure 6-36  Select Transaction Resource Usage List: File Usage](image)

When the job completed, we received the report shown in Figure 6-37.

![Figure 6-37  Usage List](image)
The Transaction Resource Usage List report provides a detailed list of transaction resource class records. The report consists of two sections:

- The Task Identification section that identifies the CICS task
- The Resource section(s) associated with the CICS task immediately above it

This report can be very useful for analyzing transaction resource usage.

6.8 Conclusion

Using CICS PA reports and extracts, we tuned our LSR pool. We also compared performances of a set of VSAM applications running in our environment when VSAM data sets were open in either LSR or RLS mode.
Tuning the CICS DB2 attachment facility

This chapter provides a brief overview of the CICS DB2 attachment facility and CICS DB2 accounting and monitoring. Then it describes the use of CICS Performance Analyzer (PA) and the reports it can produce for a CICS region running with a DB2 workload.

Two scenarios in this chapter demonstrate CICS PA reports for CICS regions running a CICS Transaction Server for OS/390 V1.3 and CICS Transaction Server for z/OS V2.2. The CICS regions are connected to DB2 V7.1 subsystem.

This chapter also discusses the system setup of the CICS regions and describes the changes provided in CICS Transaction Server for z/OS V2.2 and the Open Transaction Environment (OTE).

Note: These scenarios were used to provide situations that allow us to demonstrate the use of CICS PA reports when running with a CICS DB2 workload. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, these scenarios and the results provided are for demonstration purposes only. They do not provide definitive results for a customer environment.
7.1 CICS and DB2

This section describes functions provided by the CICS DB2 Attachment Facility. For more information, refer to the “Accounting and monitoring in a CICS DB2 environment” chapter in the CICS Transaction Server for z/OS CICS DB2 Guide, SC34-6014.

7.1.1 Overview of the CICS DB2 Attachment Facility

The CICS DB2 Attachment Facility provides a multithread connection between the two environments. The architecture allows a CICS task to access both DB2 and CICS recoverable resources with data integrity. Each CICS transaction accessing DB2 uses a different MVS task control block (TCB), thus exploiting a multiprocessor's capability for overlapped processing.

The CICS DB2 Attachment Facility enables application programs running under CICS to forward DB2 commands from CICS, by establishing a connection and a communication path between CICS and DB2 subsystems.

DB2 provides attachment facilities for MVS subsystems and batch address spaces to access DB2 resources through connections established by using the MVS Subsystem Interface (SSI) protocol. The CICS connection allows access to DB2 resources from a transaction that is also accessing DLI and VSAM resources. Before a CICS subsystem can access DB2 resources, it must establish a connection to DB2 and then create one or more threads on the connection. The connection establishes a communication path between the subsystem or address space and the DB2 subsystem.

7.1.2 Functions

The CICS DB2 attachment facility provides three major functions:

- **Application programming interface (API)**
  DB2 provides a language interface module that allows a CICS application written in Assembler, C, COBOL, PL/I, or Java to access DB2 databases by using the data manipulation language (DML) subset of SQL. It also allows you to define DB2 objects and control authority (GRANT and REVOKE) by using the data description language (DDL) subset of SQL. Updates to DB2 resources are fully synchronized with updates to CICS-protected resources such as file control, temporary storage, intrapartition transient data, and DL/I databases. The CICS Attachment Facility controls the routing of SQL statements to DB2 and the synchronization of commit processing between the two subsystems through the CICS task-related user exit (TRUE) function.

- **Attachment commands**
  Attachment commands display and control the status of the attachment facility and are issued through the supplied CICS transaction, DSNC. You can use the attachment commands to start the connection to DB2 (STRT), stop the connection to DB2 (STOP), display CICS-DB2 thread status and statistics (DISP), and modify the characteristics of the connection to DB2 (MODI). You control the use of these CICS attachment facility commands through the standard CICS security mechanisms. The commands are not routed to DB2, so there is no DB2 authorization checking.

- **DB2 commands**
  After a connection between CICS and DB2 is established, you can use the CICS-supplied transaction, DSNC, to issue DB2 commands to the DB2 system. The DB2 commands are routed to DB2 for processing. DB2 checks that the user has DB2 authority to issue the commands. The commands are used to display and control the status of the DB2 system.
All DB2 commands that you enter through CICS must start with a dash (-) to show that the command is a DB2 command rather than an attachment command.

### 7.1.3 CICS and DB2 connectivity

Two or more CICS systems can share the same DB2 subsystem. However, each CICS system can be connected to only one DB2 subsystem at the same time. When an application program operating in the CICS environment issues its first SQL request, CICS and DB2 process the request as follows:

- A language interface, or stub, DSNCLI, that is link-edited with the application program calls the CICS resource manager interface (RMI).
- The RMI processes the request, and passes control to the CICS DB2 Attachment Facility's task-related user exit (TRUE), the module that invokes DB2 for each task.
- The CICS DB2 Attachment Facility schedules a thread for the transaction. At this stage, DB2 checks authorization, and locates the correct application plan.
- DB2 takes control, and the CICS DB2 Attachment Facility waits while DB2 services the request.
- When the SQL request completes, DB2 passes the requested data back to the CICS DB2 Attachment Facility.
- CICS now regains control, and the CICS DB2 Attachment Facility passes the data and returns control to the CICS application program.

Within the connection, a thread establishes a bidirectional path between a user in a subsystem or batch address space and specific DB2 resources (application plan or command processor). Multiple threads can be established between a connected CICS and DB2. In CICS, there is a thread for each active CICS transaction accessing DB2.

The types of thread provided by the CICS DB2 attachment facility are:

- **Command threads**: These threads are reserved by the CICS DB2 attachment facility for issuing commands to DB2 using the DSNC transaction.
- **Entry threads**: These threads are specially defined threads intended for transactions with special requirements. You can instruct the CICS DB2 attachment facility to give entry threads to particular CICS transactions. DB2 entry threads can be protected. This is achieved by specifying PROTECTNUM(x) and THREADLIMIT(x) in the DB2ENTRY definition. A new protected thread is only created if an existing one is not available for reuse. A thread is protected, if at thread termination the number of protected threads is less than the PROTECTNUM value (and no new work is queued). After the thread is marked as protected, it is terminated, if it is unused for two consecutive purge cycles.
- **Pool threads**: These threads are used for all transactions and commands that are not using an entry or DB2 command thread.

Each CICS transaction that accesses DB2 needs a thread, an individual connection into DB2. Each thread runs under a thread task control block (thread TCB) that belongs to CICS. CICS and DB2 both have connection control blocks linked to the thread TCB. The nature of the thread TCBs, and the way in which they are linked to the DB2 connection control block (and therefore the thread), differs depending on the version of DB2 to which CICS is connected.

While CICS is connecting to a DB2 subsystem, it checks the DB2 release level of the subsystem. If CICS is connecting to DB2 Version 6 or later, the CICS DB2 task-related user exit (TRUE, the module that invokes DB2 for each task) is automatically enabled as open API, so it can use the CICS OTE. If CICS is connecting to DB2 Version 5 or earlier, the TRUE is not enabled as open API, and does not use OTE.
Thread TCBs in a non-Open Transaction Environment

When CICS is not using the OTE, the thread TCBs are subtasks created by the CICS DB2 Attachment Facility to run each thread that is requested by transactions or DB2 commands. The TRUE itself remains on the CICS main TCB, the QR TCB.

Figure 7-1 summarizes how thread TCBs operate in a non-Open Transaction Environment. Here we can see CICS using a thread to access DB2. The application has invoked the RMI, which invokes the CICS DB2 Attachment Facility's TRUE. The CICS DB2 TRUE, operating on the CICS main TCB, uses an assembly consisting of a subtask TCB, a CSUB, and a DB2 connection control block to run a thread into DB2. The plan associated with the thread is held in DB2. The second thread in the diagram is one that is not currently in use, but is protected.

For a CICS DB2 application to run on a thread TCB, a TCB switch is required from the CICS main TCB (QR TCB) onto the thread TCB. On the return from DB2, another switch is required back to the QR TCB. See Figure 7-2 for an example.
Figure 7-2   TCB Switching in CICS Transaction Server (TS) V1.3

Thread TCBs in Open Transaction Environment

When CICS is using the OTE, the CICS DB2 Attachment Facility uses open TCBs (L8 mode) as the thread TCBs. Open TCBs perform other tasks besides accessing DB2 resources. In the Open Transaction Environment, the CICS DB2 TRUE runs on an open TCB rather than on the CICS main TCB.

An application can also be defined as threadsafe, in which case it can continue to run on the L8 TCB after the DB2 SQL request has completed, including running EXEC CICS commands that normally run on the QR TCB. A threadsafe application must provide serialization of shared resources so they cannot be accessed by other user tasks at the same time. If the application is determined to be threadsafe, then you can define it using the CONCURRENCY(THREADSAFE) attribute on the PROGRAM definition. Note that some EXEC CICS commands are non-threadsafe.

Before the first SQL request, the application program runs on the CICS main TCB, the QR TCB. When it makes an SQL request and invokes the TRUE, control passes to the L8 TCB, and DB2 processing is carried out. On return from DB2, if the application program is threadsafe, it now continues to run on the L8 TCB, avoiding an expensive TCB switch.

Refer to the CICS Transaction Server for z/OS CICS Application Programming Reference, SC34-5994, for a more detailed explanation of threadsafe programs.

Figure 7-3 shows CICS using a thread to access DB2 in the Open Transaction Environment. The CICS DB2 TRUE was invoked by the RMI, and is operating on an open TCB. The CICS DB2 Attachment Facility has associated a CSUB and a DB2 connection control block with the open TCB. The DB2 connection control block has a thread into DB2. The plan associated with the thread is held in DB2. The diagram also shows a thread that is not currently in use, but is protected, and two open TCBs that are available for reuse.

If an application program in the OTE is not threadsafe, the CICS DB2 TRUE still runs on an L8 TCB, but the application program runs on the QR TCB throughout the task. Every time the program makes an SQL request, CICS switches from the QR TCB to the L8 TCB and back again.
Executing a CICS application which has been defined with the CONCURRENCY(THREADSAFE) attribute on the PROGRAM definition, but contains a non-threadsafe EXEC CICS command, is considered threadsafe (and continues to execute on the L8 open TCB) until the time when control is switched to the QR TCB to execute the non-threadsafe EXEC CICS command. See Figure 7-4 for an example.

To define the number of open TCBs (L8) that are available to CICS, you use the SIT parameter MAXOPENTCBS. The DB2CONN parameter, TCBLIMIT specifies how many of these will be used by DB2.
7.2 CICS DB2 accounting and monitoring

To obtain the complete picture of a CICS DB2 system environment, we need to capture storage management subsystem (SMF) records from both CICS and DB2. From CICS, we need the CMF records (SMF 110) and from DB2, we need the DB2 Accounting records (SMF 101).

The information about CPU accounting in DB2 is collected by activating DB2 Accounting trace Class 1 and Class 2. Refer to 1.3.2, “DB2 accounting data (SMF 101 records)” on page 14.

Class 1 results in accounting data being accumulated by several DB2 components. The elapsed time of a DB2 thread is included in this data. Class 2 collects the elapsed and processor times spent in DB2. It is important to remember that with CICS TS V2.2 connected to DB2 V6 or later, the DB2 Class 1 time is included in the CICS CPU time and the DB2Wait field shows zero.

Figure 7-5 shows each period of processor time that is reported by CICS and DB2 when CICS is connected to DB2 Version 5 or earlier.

![Figure 7-5: CPU accounting for DB2 Version 5 or earlier](image-url)
Figure 7-7 shows each period of processor time that is reported by CICS and DB2 when CICS is connected to DB2 Version 6 or later.

Using CICS PA to analyze the CICS performance class records, you can check the elapsed time a transaction spends waiting for a DB2 request. Also, you can check the attach overhead on the CICS side. The CICS PA fields used are:

- **RMISUSP**: The total elapsed time the task was suspended by the CICS dispatcher while in the CICS RMI.
- **DISPWAIT**: The time the task waited to resume execution.
- **RMITIME**: The amount of elapsed time spent in the RMI.

Also, in the CICS performance class, you have DB2 statistics that can be very useful:

- **DB2CONWT**: The elapsed time during which the user task waited for a CICS or DB2 subtask to become available.
- **DB2RDYQW**: The elapsed time during which the user task waited for a DB2 thread to become available.
- **DB2REQCT**: The total number of DB2 EXEC SQL and Instrumentation Facility Interface (IFI) requests issued by the user task.
7.3 CICS PA reporting CICS DB2 Attachment Facility

The CICS PA DB2 reports combine the CICS CMF performance class records (SMF 110) with the DB2 Accounting records (SMF 101) to produce a consolidated and detailed view of DB2 usage by your CICS systems. The DB2 report enables you to view CICS and DB2 resource usage statistics together in a single report.

The DB2 List report shows detailed information about DB2 activity for each transaction. The DB2 Summary reports summarize DB2 activity by transactions.

The reports can include the following DB2 information:
- DB2 Thread Identification
- Class 1 Thread elapsed and CPU times
- Class 2 In-DB2 elapsed and CPU times
- Class 3 Suspend times
- Buffer Manager statistics
- Locking statistics
- SQL DML statistics

The DB2 report matches CICS Monitoring Facility Performance records with the DB2 Accounting records by network unit of work (UOW) ID. Your CICS-DB2 resources must be defined with the DB2ENTRY attribute ACCOUNTREC(TASK) or ACCOUNTREC(UOW) for matching to occur.

CICS PA supports DB2 Accounting statistics data from DB2 Version 5, Version 6, and Version 7, although we only use DB2 Version 7 in our scenarios.

A Recap report showing processing statistics is always printed at the end.

7.4 CICS PA example reports

CICS PA provides DB2 Performance reports. These reports capture both the CMF and DB2 Accounting records. We show here several CICS PA reports which are used to demonstrate the various options available when using DB2 reports. Refer also to Figure 19-18 on page 428 and Figure 19-19 on page 429, which show using the Historical Database for DB2 reporting.

First, we needed to acquire the SMF data into CICS PA. We obtained the SMF data in data set SMFDATA.ALLRECS.G8429V00. This data set contained the data for both SMF 110 and SMF 101 records. We then used the Take-up function, by selecting option 4 (Take-up) from SMF File from the System Definition Menu. This function automatically adds any systems to CICS PA that are found within SMF records on this data set.
An example of the System Definitions screen is shown in Figure 7-7 after the take-up.

![System Definitions screen](image)

We then needed to create a report set for the DB2 reports. We requested the creation of a new Report Set DB2REPS (Figure 7-8).

![Creating a new report set](image)
As shown in Figure 7-9, we selected **DB2** to create the DB2 reports.

![Figure 7-9: Selecting DB2 for DB2 reports](image)

**Figure 7-9: Selecting DB2 for DB2 reports**
As shown in Figure 7-10, we specified a CICS APPLID and a DB2 SSID, all other options we left at the default.

Figure 7-10 Specifying the system requirements

The report produced is a short summary. When we returned to the Report Sets screen, both the Global and DB2 options were activated, this is indicated by the \textit{Yes} in the Active column following the option. From there, we enter RUN to run the report (Figure 7-11).

Figure 7-11 Submitting the DB2 report set
We could have overridden the system selection criteria that we specified earlier, as well as the date and time we wanted to start and stop reporting (Figure 7-12).

Example 7-1 shows the report options that were generated from this Report Set.

```
Example 7-1 Report Set options generated

* Report Set=DB2REPS
* Description=CICS PA Report Set
* Reports for System=SCSCPJA6
  * Image =SC66
  * Description=System added by take-up
  CICSPA IN(SMFIN001),
    APLID(SCSCPJA6),
    LINECNT(60),
    FORMAT(':'(),''),
    DB2(OUTPUT(DB2R0001),
     EXTERNAL(CPAXW001),
     SSID(D7Q2),
     SHORTSUM,
     MAXLONGSUM,
     TITLE1('CICS PA example DB2 report'))
/*
```

Figure 7-12 Run Report Set screen
The listing output from the run of these report options is shown in Figure 7-13.

In the DB2 Short Summary report, for each APPL ID, a data line is presented for the CMF performance class data summarized by transaction and program, and a data line is presented for the associated DB2 Accounting data summarized by the SSID and plan name.

We repeat the same exercise, but this time we are going to request the Long Summary report from the DB2 Report Set screen as shown in Figure 7-14.
Figure 7-15 shows the report the Long Summary option provides.

![Long Summary report](image)

A DB2 report list was then produced. To produce a report list, we needed to choose the **List** option from the DB2 Report screen as shown in Figure 7-16.

![Requesting the List report](image)

### File Systems Options Help

```
DB2REPS - DB2 Report

Command ===>  More: +

CICS System Selection:  Report Output:
APPLID . . SCSCPJA6 +  D0name . . . . . . . . . . . . . DB2R0001
Image . . SC66 +  Print Lines per Page . . (1-255)
Group . . +

DB2 System Selection:  Report Options:
SSID . . . D7Q2 +  / Process DB2 Accounting records
Image . . SC66 +  List records with no DB2 activity
Group . . +  / Long Summary with DB2 maximums

Reports: ----- DB2 Accounting data to include in report -----
Required:  Class1 Class2 Class3 Buffer Locking DML 1 DML 2
/ List / / / /
Long Summary / / / /
Short Summary

Report Format:
Title . . CICS PA example DB2 report

Selection Criteria:
Performance
```
The DB2 List report provided a detailed list of all network UOWs with DB2 activity. This report consolidates CICS CMF performance class records and DB2 accounting statistics from a single or multiple CICS systems. Figure 7-17 shows an example of this report.

In the DB2 List report, a data line is presented for each CMF performance class record, and a block of data lines is presented for each associated DB2 Accounting record. Records that are part of the same network UOW are sequentially in groups separated by blank lines. A network UOW will only be presented if it involved some DB2 activity.

A Recap report showing processing statistics is always printed at the end (Figure 7-18). The recap statistics that are shown can be useful in seeing where the records that were processed came from.
If the DB2 Accounting records have a large number of excluded records, then it is a good indication that the DB2ENTRY definitions do not have ACCOUNTREC(TASK) or (UOW) set.

### 7.5 CICS TS V1.3 and DB2 V7 scenario

This scenario shows how you can use CICS PA to produce several DB2 reports from SMF data for CICS TS V1.3 and DB2 V7.1 systems.

DB2N transaction performs the same 15 fetches from the DSN8710.EMP table, but between each fetch there is a non-threadsafe EXEC CICS command. In this scenario, we use the reports CICS PA generates to monitor the number of times the transaction DB2N overflows to the POOL.

The workload was generated using Teleprocessing Network Simulator (TPNS). We used TPNS to simulate a realistic 3270 workload environment. The SIT parameter, TCBLIMIT is 12 and the DB2ENTRY parameter, THREADLIMIT for DB2N is 10.
To start, we added the CICS TS V1.3 CICS system. We already added the DB2 system using the Take-up command in the example reports chapter. The name of this CICS system is SCSCPAA1. To add the new system we had a number of ways of doing this. In this case, we went to the System Definitions menu and enter I (insert) to add a new line (Figure 7-19).

![Figure 7-19 Inserting a new system](image)

We were then presented with a pop-up screen to add the new system; we entered the CICS system name and then a 1 to indicate that it is a CICS system. The next screen was then displayed. Here we added the SMF data set we wanted to use to obtain the performance records (Figure 7-20).

![Figure 7-20 Adding SMF data sets](image)

We then changed the DB2 system definition, to change the SMF data sets it used. To do this, we selected D7Q2 from the Systems Definitions menu and then added the same SMF data sets that we had added for the CICS system.

We then used the Report Set DB2REPS that we set up earlier.
For this scenario, we put the CICS system and the DB2 system into a group. For this, we selected option 3 on the Systems Definition menu. We set up a new group by specifying **DB2GROUP**, we then needed to add the systems we were interested in obtaining performance records for. Figure 7-21 shows the group that contains our CICS system and our DB2 system.

--- System Definitions ---

**Command ==>**

**Group . . . . . . DB2GROUP**

**Description ...**

/  System + Type     Image                 Description
SCSCPAA1  CICS              CICS TS V1.3
D7Q2      DB2      SC66     System added by take-up

*End of list*

--- System Selection ---

/ Exc APPLID + Image + Group + Output Criteria
DB2GROUP DB2R0001 NO

*End of list*

--- Using groups ---

We then went to the Report Set DB2REPS and added the group (Figure 7-22).
The first report we wanted to run is the short summary as shown in Figure 7-23.

![Figure 7-23 Request Short Summary report](image)

<table>
<thead>
<tr>
<th>Command</th>
<th>More:</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS System Selection:</td>
<td>Report Output:</td>
<td></td>
</tr>
<tr>
<td>APPLID . +</td>
<td>DDname . . . . . . . . . DB2R0001</td>
<td></td>
</tr>
<tr>
<td>Image . +</td>
<td>Print Lines per Page .. (1-255)</td>
<td></td>
</tr>
<tr>
<td>Group .. DB2GROUP +</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB2 System Selection:</th>
<th>Report Options:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID . . +</td>
<td>/ Process DB2 Accounting records</td>
<td></td>
</tr>
<tr>
<td>Image . +</td>
<td>List records with no DB2 activity</td>
<td></td>
</tr>
<tr>
<td>Group . . +</td>
<td>/ Long Summary with DB2 maximums</td>
<td></td>
</tr>
</tbody>
</table>

Reports ----- DB2 Accounting data to include in report -----
Required: Class1 Class2 Class3 Buffer Locking DML 1 DML 2
List / / / / |
Long Summary / / / / |
/ Short Summary |

Report Format: |
Title . CICS PA example DB2 report |

Selection Criteria: |
Performance |
We were then ready to RUN the Report Set. You can do this in several places. Here we entered the RUN command from the Report Set as shown in Figure 7-24.

![Figure 7-24 Submitting the report set](attachment:7-24.png)
We then received another screen which allowed the changing of various options, such as the systems we wish to collect performance data from, the option to edit the JCL that will be submitted or specify a date and time we want to collect the performance data. What we requested is shown in Figure 7-25.

![Figure 7-25 Selecting records by date and time]

We then looked at the output, shown in Figure 7-26, from the Report Set.

![Figure 7-26 Short Summary report of the DB2N transaction]

Here, we had an average response time of .0581 and the CICS PA field DB2ConWt of .0040. The DB2ConWt field indicates the DB2 Connection Wait time, the wait for a DB2 subtask to become available. We then ran the DB2 Report Set with the Long Summary option. We also requested DB2 Accounting data to be included in the report. The data we requested is the Class 1, Class 2, Buffer and Locking, as shown in Figure 7-27.
We then looked at the Long Summary report listing, as shown in Figure 7-28. We overflowed to the pool 2089 times.

In conclusion, we could deduce from the CICS PA long summary report that we should think about increasing the THREADLIMIT parameter for the DB2ENTRY for DB2N, in an effort to reduce the number of pool overflows.
7.6 CICS TS V2.2 and DB2 V7 scenario

The goal of this scenario is to show the difference in system resources used by a non-threadsafe and a threadsafe application running in CICS TS V2.2 and connected to DB2 V7.1. The CICS PA is used to produce reports using the various options available to the DB2 Report Set.

To show the difference in a threadsafe and a non-threadsafe environment, we use two simple transactions.

- The DB2R transaction which performs 15 fetches from DSN8710.EMP, the sample employee table supplied with DB2 V7.1. This application is threadsafe and was defined as such, with the program attribute, CONCURRENCY(THREADSAFE).

- The DB2N performs the same 15 fetches from DSN8710.EMP, but between each fetch there is a non-threadsafe EXEC CICS command. This application is non-threadsafe and was defined with the program attribute, CONCURRENCY(QUASIRENT).

The workload was generated using Teleprocessing Network Simulator (TPNS). We used TPNS to simulate a realistic 3270 workload environment. Two separate runs were carried out, one with just DB2R transactions running, and one with just DB2N transactions running.

Running the non-threadsafe scenario

The SIT parameter MAXOPENTCBS value was 15, the DB2CONN TCBLIMIT parameter value was 12 and the DB2ENTRY attribute, THREADLIMIT for DB2N was 10.

For this scenario, we set up another CICS PA group selecting option 3 from the System Definitions Menu. On the Group screen, we requested a new group called db2grp as shown in Figure 7-29.

![Figure 7-29 Groups](image-url)
After the new group was created, we added the systems that we were interested in. Here we were interested in SCSCPJA6, a CICS TS V2.2 system, and D7Q2, a DB2 V7.1 system. The screen shown in Figure 7-30 shows the two systems added to our new group.

---

File Edit Options Help

Systems in this Group Row 1 to 2 of 2

Command ===> Scroll ===> CSR

Group . . . . . . DB2GRP
Description . . .

/ System + Type Image Description
SCSCPJA6 CICS SC66 System added by take-up
D7Q2 DB2 SC66 System added by take-up

End of list

---

**Figure 7-30 Systems added to group**

To select the transactions we wanted, DB2N and DB2R, we used an Object List. We needed to create one first. We did this by accessing option of the main screen. We then specified new DB2TRAN on the Objects Lists screen. The screen in Figure 7-31 was then displayed.

---

File Edit Confirm Options Help

EDIT Object List - DB2TRAN Row 1 to 1 of 1

Command ===> Scroll ===> CSR

Description . . . . CICS PA Object List

Specify the Object List values:

/ 1st Value 2nd Value Sublist
DB2*

End of list

---

**Figure 7-31 Object List**
On this screen, we entered a value of DB2*. We could use the Object List as the selection criteria for a Report Set. First we ran a short summary report as shown in Figure 7-32. We also selected the Performance selection criteria.

![Figure 7-32 Request Short Summary report](image)

On the Performance Selection Statement screen (Figure 7-33), we specified an Object List of DB2TRAN.

![Figure 7-33 Performance Select Statement screen](image)

We then submitted the Report Set. Part of the listing of the short summary is shown in Figure 7-34. From this report, we could see that the average response time for the DB2N transaction was 0.0410, and that we had an average wait for an open TCB to become available.
of .0020. This indicates that maybe we should increase the value of the DB2CONN parameter, TCBLIMIT.

Figure 7-34  Short Summary report of DB2N transaction

We then created a Long Summary report (Figure 7-35) to provide more details about the DB2N transaction. From the Long Summary report, we could see the maximum time our transaction waited to obtain an open TCB. We could also see that we had overflowed into the pool 953 times, showing that we should think about increasing the THREADLIMIT value for the DB2N transaction, if this overflowing to the pool is not expected.

Figure 7-35  Long Summary report of DB2N transaction

The DB2 Accounting record Class1: Thread Time is now included in the CICS CPU time. Therefore, the CICS PA field, UserCPU time will be greater than or equal to the Class 1 time.

Finally, we produced a DB2 List report. The List report shows a detailed list of all network UOWs with DB2 activity (Figure 7-36).
We changed the SIT parameter, MAXOPENTCBS to 20, the DB2CONN parameter, TCBLIMIT to 20 and the DB2ENTRY attribute, THREADLIMIT for DB2N to 20. We ran the tests again and collected the SMF records. We used the Long Summary report to display the results of this new test (Figure 7-37).

From this report, we could see that we had reduced the DB2ConWt to 0 and the number of overflows to the pool to 0.
Running the threadsafe scenario
We then ran the DB2R transaction with the same system setup. The SIT parameter, MAXOPENTCBS parameter was 15, the DB2CONN parameter, TCBLIMIT was 12 and the DB2ENTRY attribute, THREADLIMIT for DB2R was 10.

The Short Summary report is shown in Figure 7-38.

![Figure 7-38 Short Summary report of the DB2R transaction](image)

Looking at this report, we could see that the average response time is .0150. Even though we used the same number of open TCBs, we had a much smaller value for the CICS PA field, DB2ConWt. We then looked at the Long Summary report as shown in Figure 7-39.

![Figure 7-39 Long Summary report of DB2R transaction](image)

The number of transactions that overflowed to the pool were reduced without the need to change any other parameters.

In conclusion, we could see the improvement in CICS DB2 performance and system resources when running in a threadsafe environment quickly with the use of the CICS PA DB2 reports.
7.7 Extracting CICS DB2 records

To extract CICS DB2 performance records, we need to define a Report Form. We selected option 3 on the Primary Options Menu, and then arrived at the Report Forms screen. As shown in Figure 7-40, we created a new Report Form called \textit{db2perf}.

We were then given the option of which APPLIDs we are interested in extracting. We requested an APPLID of \texttt{SCSCPAA1} and a form type of \texttt{List} (Figure 7-41).

The List Report Form screen (Figure 7-42) was displayed. From here, we chose which fields we wanted to extract. Since we were interested in CICS DB2 performance, we needed to move the DB2 fields we had available before the EOX line.
Figure 7-42  List Report Form screen

We created another report set. Here we chose Export. The next screen showed the Exports screen, where we specified the Form we just created (Figure 7-42), along with the data set we were using to export the data (Figure 7-43).

Figure 7-43  Exports screen

We were only interested in the CICS transactions DB2N and DB2R, so we selected them in the Performance select statement as shown in Figure 7-44.
We then submitted this Report Set and then viewed the job output. The extract informed us of how many records we had extracted. In Figure 7-45, you see that we extracted 11,855 records.

![Figure 7-44 Performance Select Statement screen](image)

We then viewed the output data set using ISPF. Figure 7-46 shows the output.

![Figure 7-45 Record count of extracted records](image)

We then transferred the data set down to a PC using the Personal Communications emulation program file transfer.

![Figure 7-46 Extract of CICS DB2 data](image)
The transfer options we used in Personal Communications were the name of the host file and the name for the PC file that should be placed in the appropriate directory and should be called .TXT. The transfer type was text. We imported the data into Microsoft Excel.

We started Microsoft Excel and clicked **File -> Open**. We selected the file we wanted to open. Then the Text Import Wizard screen opened. Here, we chose **Delimited** and then **Semicolon**. Semicolon was the delimiter we used when creating the extract file in CICS PA.

After we had the data in Microsoft Excel, we created a chart. As shown in Figure 7-47, we produced a graph that shows the average CPU Time for the DB2N and the DB2R transactions we ran earlier. It also shows the improvement in CPU times for DB2R over DB2N.

![Figure 7-47 CPU Time by transaction](image)

### 7.8 Conclusion

Using a set of CICS PA reports and extracts, we showed the performance advantages of using the threadsafe environment as opposed to a non-threadsafe. We also showed how you can tune the MAXOPENTCBS SIT parameter, TCBLIMIT DB2CONN parameter, and THREADLIMIT DB2Entry parameter to improve overall performance or the performance of a specific transaction.
WebSphere MQ

With the new WebSphere MQ report, CICS Performance Analyzer (PA) 1.3 offers the possibility to produce comprehensive reports containing detailed information about the MQ application programming interface (API) calls that are executed in CICS transactions. This chapter provides an overview of the new reports that can be produced from the MQ storage management subsystem (SMF) 116 accounting records written only from CICS transactions.

First we give a brief overview of MQ accounting trace records and describe the MQ provided sample transactions that we used. Then we go through a scenario. We show the content of MQ list and summary reports and how CICS performance data can be related to MQ reports. We also provide a sample of a Transaction Temporary Storage Usage Summary report.
8.1 Overview

CICS PA MQ reports use the WebSphere MQ accounting data included in an SMF type 116 record to provide a performance analysis of the CICS transactions that access an MQ queue manager. WebSphere MQ accounting records are produced when the Accounting Trace component of WebSphere MQ is activated. MQ traces can be selected by class. There are two possible classes for an MQ accounting record:

- **Class 1** records contain the CPU time spent processing WebSphere MQ API calls and the count of MQGET and MQPUT calls.
- **Class 3** records contain enhanced accounting and statistical data for each task at thread and queue level.

There are two types of CICS PA MQ reports:

- **CICS PA MQ List report**: This report provides a detailed trace of the WebSphere MQ accounting records, reporting the comprehensive performance data contained in the Class 1 and Class 3 records.
- **CICS PA MQ Summary report**: This report provides an analysis of the MQ system and queue resources used and the transactions they service. The data can be summarized by CICS transaction ID or MQ queue name or both.

CICS PA 1.3 supports the WebSphere Accounting and statistical data from MQSeries for OS/390 Version 5.2 and from IBM WebSphere MQ for z/OS Version 5.3 and 5.3.1.

8.2 Environment

We started CICS region CICSLSA5 that connects to a WebSphere MQ queue manager. The SSID of the MQ subsystem is MQFI. The name of the queue that we used is CICSPA.MQ. We did not run a workload but decided to use the MQ provided sample transactions to see the result of the execution of these transactions in the CICS PA MQ reports.

The sample MQ transactions are:

- **MVPT** executes program CSQ4CVK1 that is a sample program to put a number of messages to a queue.
- **MVGT** executes program CSQ4CVJ1 that is a sample program to get a number of messages from a queue. These messages are written to a CICS temporary storage queue.

The transaction syntax is:

```
MVPT, nummsgs, padchar, msglength, persistence, qname
MVGT, nummsgs, gettype, syncpoint, qname
```

Note the following explanation:

- **nummsgs** is the number of messages written to or read from the queue.
- **padchar** is the character that will be written to the message buffer.
- **msglength** is the length of the message.
- **persistence** is P for a persistent message or N for a non-persistent message.
- **qname** is the name of the queue.
- **gettype** is B for a BROWSE-GET or D for a DESTRUCTIVE-GET.
- **syncpoint** is S for a SYNCPOINT or N for NO-SYNCPOINT.
8.3 CICS changes

Authorized program analysis report (APAR) PQ76703 adds some new monitoring and statistics functions to CICS TS 2.2. Refer to the list of enabling PTFs in Table 2-1 on page 26. One of the new options specifies whether you want additional monitoring performance class data to be collected for the resource managers used by your transactions. This way, the time spent in the External Resource Managers (ERM) is added in different new Resource Manager Interface (RMI) fields in the CMF Performance class record. This option is activated via a new RMI parameter on the TYPE=INITIAL macro of the MCT.

Before running this scenario, we assembled the MCT that is shown in Example 8-1. It shows the specification of the RMI=YES parameter and the suffix that we specified for the MCT parameter in the SIT or SYSIN overriding at CICS initialization.

Example 8-1  MCT with RMI=YES

```plaintext
// JOB /*JOBPARM SYSAFF=SC66
//PLEASE EXEC DFHAUPLE,INDEX='CICSTS22.CICS',
//       INDEX2='CICSTS22.CICS'
//ASSEM.SYSUT1 DD *
RMI      DFHMCT TYPE=INITIAL,                                          *
        APPLNAME=YES,                                           *
        FILE=10,                                                *
        RMI=YES,                                                *
        SUFFIX=RM
        *
        DFHMCT TYPE=FINAL
        END
//LNKEDT.SYSLMOD DD DISP=SHR,DSN=CICSSYSF.APPL62.LOADLIB
//LNKEDT.SYSIN DD *
        NAME DFHMCTRMRM(R)
```

This scenario includes the MQ RMI field in the reports that we print with CICS PA.

8.4 MQ accounting trace

Before we ran the sample MQ transactions, we started the MQ traces by entering the command:

- MQFI START TRACE(ACCTG) DEST(SMF) CLASS(01:03)

From a CICS terminal, we ran the following transactions:

- MVPT,500,*40,P,CICSPA.MQ
- MVPT,100,*400,N,CICSPA.MQ
- MVGT,100,B,N,CICSPA.MQ
- MVGT,100,D,N,CICSPA.MQ
- MVPT,50,*50,P,CICSPA.MQ
- MVPT,50,*1400,P,CICSPA.MQ
- MVGT,100,B,N,CICSPA.MQ
- MVGT,100,D,N,CICSPA.MQ
- MVGT,100,D,N,CICSPA.MQ
- MVGT,100,D,N,CICSPA.MQ
On the run of the last transaction, the MVGT transaction returned following message on the screen:

MQGET 000000101 failed * CC : 000000002 * RC : 000002033 *

We received a return code of 2033 indicating that no messages were available.

We switched the SMF data sets and copied the CICS and MQ records to a separate data set called BARI.SMFDATA.MQ4. We used this data set as input for the CICS PA Take-up function. From the CICS PA Primary Option Menu, we selected option 1 (System Definitions), and then option 4 (Take-up). Next, we entered the data set name as shown in Figure 8-1.

![Figure 8-1 System Definitions Take-up function](image)

We returned to the CICS PA Primary Option Menu and selected again the System Definitions option. A screen is displayed indicating that the system definitions were updated by the Take-up function. Figure 8-2 shows the System Definitions screen.
8.5 MQ reports

We were ready to produce our first MQ report. On the CICS PA Primary Option Menu, we selected Report Sets. On the Report Sets screen, we entered NEW MQ to create a new report set with the name MQ. Figure 8-3 shows the screen that is displayed.

---

**Figure 8-2** MQFI and SCSLSA5 were added to the system definitions

**Figure 8-3** New report set with name MQ
8.5.1 MQ class 1 reports

On the Report Set screen, we selected WebSphere MQ in the Subsystem Reports category resulting in the screen shown in Figure 8-4.

We specified SSID, chose both a List and a Summary report with the summary sorted by transaction name, and entered the queue name for which we wanted the reports. Note that the queue name is case sensitive.

When we pressed Enter, the Image field was automatically updated with SC66. We entered a meaningful report title and selected selection criteria. Figure 8-5 shows our select statement. We chose to only look at the sample MQ transaction identifiers starting with MV.
We returned to the Report Set panel where we entered the RUN line action on the WebSphere MQ report. The resulting reports are shown in Figure 8-6.

This report lists the transactions that we executed. Tasks 55 and 56 are adding respectively 500 and 100 messages of 40 and 400 bytes to the queue making a total of 600 messages on the queue. Task 57 reads 100 messages. However, in a Class 1 report, we cannot see that these are non-destructive browse operations. The Class 3 report allows us to distinguish between a browse and a destructive get. Task 58 also does 100 reads but these are destructive, bringing the number of messages to 500. Tasks 59 and 60 add each 50 messages, also with different lengths. The queue depth again is 600 messages. Tasks 61, 62, 63, and 64 each read 100 messages, thus removing the 400 remaining messages with a length of 40. Task 65 also reads 100 messages and removes the 100 messages with a length of 400 bytes. Task 66 asked to get 102 messages but only 100 were left, 50 with length of 50 and 50 with length 1400. After these were read, we received the reason code of 2033 telling that there were no more messages.

The Class 1 Summary report gives an overview per transaction code. It calculates an average value for all fields.

For a more detailed look at the MQ activity, we requested a Class 3 List report. This produces a list of extended information per transaction. The Class 3 report contains information in five sections. For detailed information about these five sections and the fields that these contain, see “WebSphere MQ report” in the CICS Performance Analyzer for z/OS Report Reference, SC34-6308.
Being interested in tasks 55, 57, and 66, we changed the selection criteria before requesting a Class 3 report. Figure 8-7 shows our Performance select statement where we added the three task numbers.

**Figure 8-7  Updated selection criteria for selected tasks**
8.5.2 MQ Class 3 reports

We then changed the WebSphere MQ report options to request a Class 3 List report sorted by transaction. Figure 8-8 shows this.

![MQ Class 3 report options](image)

The resulting report is shown in Figure 8-9.

Task 55 shows the detail about an MVPT transaction. Tasks 57 and 66 are MVGT transactions. In this Class 3 report, we see for task 57 that after the GET count, there is a breakdown of GET request sub-types. Here we see that all GET requests were browse any requests. Other possibilities are browse specific, destructive any and destructive specific. They only appear if the count is non-zero. For task 66, we see that all GET requests were destructive.

For task 66 we also see that the GET count here is 101. We requested to read 102 messages. The Class 1 record tells that 100 messages were read, the Class 3 record tells that 101 GET requests were issued. Unfortunately, the Class 3 data does not include the reason code for an API request so we cannot see from this report that the last GET returned a reason code of 2033.
We removed the selection for the three transactions before selecting a summary report. Figure 8-10 shows that we asked for a Class 3 Summary report sorted first by queue and then by transaction.

<table>
<thead>
<tr>
<th>SSID: MQFI APPLID: SCSCLSAS Tran: MVPT Task: 55 UserID: CICSUSER NetName: N/A</th>
<th>UOWID: N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Total Calls 1 Avg Elapsed 0.04865 Avg CPU 0.000121</td>
<td>Start: 10/17/2003 11:45:55.29</td>
</tr>
</tbody>
</table>

Queue: CICSPA.MQ

<table>
<thead>
<tr>
<th>Count</th>
<th>Elapsed</th>
<th>CPU</th>
<th>Susp Elp</th>
<th>JnlWrt Elp</th>
<th>PS Req's</th>
<th>PS Rd Elp</th>
<th>Expired</th>
<th>Page Skip</th>
<th>Msgs Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN 1</td>
<td>0.000151</td>
<td>0.000149</td>
<td>0.000139</td>
<td>0.000000</td>
<td>0.000121</td>
<td>0.000121</td>
<td>0.000121</td>
<td>0.000121</td>
<td>0.000121</td>
</tr>
<tr>
<td>CLOSE 1</td>
<td>0.000102</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
<td>0.000050</td>
</tr>
<tr>
<td>PUT 500</td>
<td>0.000154</td>
<td>0.000135</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

PUT Total Bytes 20,000 #PUT w/Data 500 Min Msg Size 40 Max Msg Siz 40

<table>
<thead>
<tr>
<th>SSID: MQFI APPLID: SCSCLSAS Tran: MVGT Task: 57 UserID: CICSUSER NetName: N/A</th>
<th>UOWID: N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Total Calls 1 Avg Elapsed 0.000214 Avg CPU 0.000103</td>
<td>Start: 10/17/2003 11:46:41.39</td>
</tr>
</tbody>
</table>

Queue: CICSPA.MQ

<table>
<thead>
<tr>
<th>Count</th>
<th>Elapsed</th>
<th>CPU</th>
<th>Susp Elp</th>
<th>JnlWrt Elp</th>
<th>PS Req's</th>
<th>PS Rd Elp</th>
<th>Expired</th>
<th>Page Skip</th>
<th>Msgs Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN 1</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
<td>0.000168</td>
</tr>
<tr>
<td>CLOSE 1</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
<td>0.000062</td>
</tr>
<tr>
<td>GET 100</td>
<td>0.000140</td>
<td>0.000115</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>BRW ANY 100</td>
<td>0.000140</td>
<td>0.000115</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

GET Total Bytes 4,000 #GET w/Data 100 Min Msg Size 40 Max Msg Siz 40

<table>
<thead>
<tr>
<th>SSID: MQFI APPLID: SCSCLSAS Tran: MVGT Task: 66 UserID: CICSUSER NetName: N/A</th>
<th>UOWID: N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Total Calls 1 Avg Elapsed 0.000137 Avg CPU 0.000063</td>
<td>Start: 10/17/2003 11:49:11.90</td>
</tr>
</tbody>
</table>

Queue: CICSPA.MQ

<table>
<thead>
<tr>
<th>Count</th>
<th>Elapsed</th>
<th>CPU</th>
<th>Susp Elp</th>
<th>JnlWrt Elp</th>
<th>PS Req's</th>
<th>PS Rd Elp</th>
<th>Expired</th>
<th>Page Skip</th>
<th>Msgs Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN 1</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
<td>0.000169</td>
</tr>
<tr>
<td>CLOSE 1</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
<td>0.000042</td>
</tr>
<tr>
<td>GET 101</td>
<td>0.005375</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
</tr>
<tr>
<td>DES ANY 101</td>
<td>0.005375</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
<td>0.009894</td>
<td>0.005147</td>
</tr>
</tbody>
</table>

GET Total Bytes 72,500 #GET w/Data 100 Min Msg Size 50 Max Msg Siz 1,400
Figure 8-10   Request for Class 3 Summary report

The resulting report is shown in Figure 8-11. The first part of the report gives calculated averages about the total activity on queue CICSPA.MQ, in this case for 12 tasks. The second part of the report gives a task identification and a summary of the task-related statistics.

![WebSphere MQ Report](image)

Figure 8-11   Class 3 Summary report sorted by transaction and by queue
8.5.3 Performance List report

The CICS PA MQ reports give CICS task numbers. With this information we can list the corresponding CICS transaction information and look for the CICS resources that were used by these transactions. In our case, we can use a CICS PA List report to see the temporary storage usage of the MQ sample transactions.

We created a new report form that contains the fields that we were interested in for the analysis of the MVPT and MVGT transactions. This report form in Figure 8-12 shows that we selected to have the TIME and COUNT values of the RMIMQM field. It also shows that we moved the field that contains the number of TS requests to auxiliary storage before the end of report indicator.

![Figure 8-12](image)

In the MQ Report Set, we deactivated the WebSphere MQ Subsystem report and selected the List Performance report. The same way as we did for the WebSphere MQ report, we selected to have a report only for the transactions with an ID starting with MV.

Figure 8-13 shows the report that we obtained. The RMI MQ time field gives an approximation of how long the tasks were in the RMI for executing the MQ calls. This is not exactly the suspend time as some RMI code is executed before the MQ timer starts. This means that the RMI MQ time contains a small part of dispatch time plus suspend time.

We made the following observations:

- The RMI count for MVPT transactions is three more than the number of PUT requests. This is due to issuing three MQ API calls for OPEN, CLOSE, and sync point. For the MVGT transactions, it is two more than the number of GET requests, because no sync point is issued.
- For task 66, we know that the application was to read up to 102 messages from the queue. One hundred were read successfully and message 101 gave reason code 2033 for no...
messages available. Task 66 entered the RMI once more than the other MVGT tasks which successfully read their requested 100 messages.

- The TSPUTAux column shows that each message read was written to auxiliary temporary storage. The additional TS request that appears in TS Total is a DELETEQ TS.

Tip: We showed how to relate CICS SMF record information to MQ information based on transaction IDs. This was easy to do since we were looking at a limited number of transactions. This can be less evident when recording over long intervals where multiple different transactions can have the same task number. In this case, we recommend that you also compare the time stamp of both records.

Figure 8-13   Performance List report for MVPT and MVGT
8.5.4 Transaction Resource Usage Temporary Storage report

We also asked for a summary of the Temporary Storage usage of our transactions. To do so, we selected Temporary Storage Usage Summary in the category Transaction Resource Usage Reports as shown in Figure 8-14.

![Figure 8-14 Selecting Temporary Storage Usage Summary](image-url)
Two reports can be requested. We first asked for the Transaction Temporary Storage Usage report that summarizes Temporary Storage usage by transaction ID. Figure 8-15 shows the Temporary Storage Summary Report where we selected the Transaction Temporary Storage Usage report.

![Figure 8-15 Selecting Transaction Temporary Storage Usage](image-url)
Notice that there is no asterisk after the Performance selection Criteria. Indeed, we removed the selection for having only those transaction codes that end on MV. We expected not to see much Temporary Storage activity on this system. We simply wanted to see if there were other users of TS. The report is shown in Figure 8-16.

This report shows that two CICS transactions were also using Temporary Storage. The user transaction that we find in the list is MVGT. For the MVGT tasks, we see a breakdown that contains the TS queue name which is also MVGT. The queue name for the CICS transactions is not available because the TS requests are not done via an EXEC CICS command.

---

**Figure 8-16  Transaction Temporary Storage Usage Summary report**

We also requested for a Temporary Storage Usage report, shown in Figure 8-17.

---

**Figure 8-17  Temporary Storage Usage Summary report**

Since only one Temporary Storage queue was name found, we received a summary for the MVGT queue. This report is a combination or the two parts of the previous report.
8.6 Conclusion

This chapter showed how easily you can produce list and summary reports by using the new MQ feature in CICS PA 1.3. Using two MQ sample provided transactions, we showed how you can interpret information about MQ API calls in these reports. CICS PA brings an added value to the MQ SMF accounting record by calculating average values and providing this information in the summary report.

We also showed how you can link a performance list report to the MQ reports. This report showed the RMI MQ field that is available in CICS TS 2.2 and that is supported by CICS PA.
**CICS and MVS System Logger**

This chapter describes the interface between CICS Transaction Server (TS) and the MVS System Logger. It also introduces the System Logger reports that you can generate using CICS Performance Analyzer (PA).

**Note:** This scenario was used to provide a situation that allowed us to demonstrate the use of CICS PA System Logger reports. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, these scenarios and the results provided are for demonstration only. They do not provide definitive results for a customer environment.
9.1 CICS TS and the MVS System Logger

CICS uses the MVS System Logger for all its logging and journaling requirements. The CICS system log is used for:

- Dynamic transaction backout
- Warm and emergency restarts
- Cold starts, but only if the log contains information required for resynchronizing in-doubt units of work
- Forward recovery logs, auto-journals, and user journals

The MVS System Logger provides a programming interface to access records on a log stream.

Three hardware options are available that CICS can use:

- Non-volatile coupling facility, where logstream data is duplexed in the MVS logger data space
- Volatile coupling facility, where logstream data is duplexed to a staging data set
- Direct access storage device (DASD)-only, where logstream data is duplexed in the z/OS logger data space

**Coupling facility and DASD-only log streams**

Each log stream is a sequence of blocks of user data that the MVS System Logger internally partitions over three types of storage:

- **Primary storage**
  
  This is a structure within a coupling facility that holds the most recent records written to the log stream. Log data written to the coupling facility is also copied to either a data space or a staging data set.
  
  For DASD-only log streams, a log structure is not available. The primary medium for DASD-only logging is the staging data set. Log data written to a DASD-only log stream is held in a data space and in a staging data set. A staging data set must be defined.

- **Secondary storage**
  
  When the primary storage structure for a log stream becomes full, the older records automatically spill into secondary storage, which consists of data sets managed by SMS. This process is known as DASD offloading. The allocation of new logger data sets for DASD offloading is known as a DASD shift. For DASD-only logging, the primary storage is the staging data set. Therefore, if the staging data set fills, offloading is done in the same way as for coupling facility logging. After data is offloaded, it is still available to the MVS System Logger.

- **Tertiary storage**
  
  This storage is used as specified in your HSM policy, by which older records are migrated to some form of archive storage. This archive storage can be either DASD data sets or tape volumes.

Log data is considered “hardened” when it is written to both the coupling facility log structure and a buffer held in a data space (or to staging data sets). MVS keeps the second copy of the data for recovery in the event of a structure failure. A staging data set is always used for DASD-only logging.

Figure 9-1 shows the components of a DASD-only system.
DASD-only log streams do not use the coupling facility storage. For DASD-only log streams, the log blocks span storage buffers and DASD log data sets. A DASD-only log stream has a single-system scope. Only one system at a time can connect to a DASD-only log stream. Multiple applications from the same system can, however, simultaneously connect to a DASD-only log stream. When a System Logger application writes a log block to a DASD-only log stream, the System Logger writes it first to the local storage buffers for the system and then automatically duplexes it to a DASD staging data set associated with the log stream.

If the staging data set fills up to its defined \text{HIGHOFFLOAD} value, the System Logger begins the offload process. As a log stream fills offload data sets on DASD, the System Logger automatically allocates a new offload data set for the log stream.

When you size the log stream for use of CICS logger (DFHLOG), it is important to minimize the amount of data that is offloaded to secondary storage. The logger begins the offload process when the high offload threshold (\text{HIGHOFFLOAD}) of the log stream is reached. The offload process consists of two steps:

1. The \text{z/OS} logger physically deletes the data in the log stream that is logically marked for deletion by the CICS log-tail deletion process. This happens at activity keypoint time.

2. The \text{z/OS} logger calculates how much data must be offloaded to secondary storage, based on the difference between \text{HIGHOFFLOAD} and \text{LOWOFFLOAD}, less the amount of data that has been deleted since the last offload event. It may happen that an offload does not occur at all. This is possible when we fall below the \text{LOWOFFLOAD} value.

The CICS system log is implemented as two MVS System Logger log streams. One log stream is the primary system log stream, DFHLOG, which holds data for most normal (short-lived) in-flight units of work (UOWs). The other log stream is the secondary system log.
stream, DFHSHUNT, which holds information for UOWs that are not short-lived. These typically are UOWs that cannot complete because of backout failures, or because they are designed as long-running tasks that issue infrequent syncpoints.

9.2 CICS PA reporting on the System Logger

CICS PA processes System Logger (SMF 88) records to provide information about the System Logger log streams and coupling facility structures that are used by CICS Transaction Server for logging, recovery, and backout operations. The report can help to measure the effects of tuning changes and identify log stream or coupling facility structure performance problems.

The System Logger List report shows information about logstream writes, deletes, and events, as well as Structure Alter events for each SMF recording interval. The System Logger Summary report summarizes logstream and structure statistics so you can measure the System Logger performance over a longer period of time. These reports, when used in conjunction with the CICS logger reports produced from the standard CICS statistics reporting utilities, provide a comprehensive analysis of the logstream activity for all your CICS systems.

9.3 Scenario description

The System Logger scenario looks at several CICS and System Logger parameters that can be tuned. It uses the CICS PA System Logger reports to show the way these parameters effect the System Logger.

For this scenario, we work with the CICS system log, DFHLOG, defined to the MVS System Logger as DASD-only. The CICS region we used for this scenario was at the CICS TS V1.3 level.

We use the CICS PA logger report function to produce reports of the logger activity, and then use the reports to help tune the logger system to perform more efficiently. To provide the System Logger SMF records, we use the same application that we used in the VSAM scenario. Again this application uses TPNS to generate its workload. Refer to Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139, for a description of our VSAM scenario.

We use the SMF (Type 88) records and the CICS PA reports to monitor the effect that several parameters have on the way the log stream for DFHLOG is effected when they are changed. We look at the SIT parameter AKPFREQ, and the logstream definition parameters LS_SIZE (logstream offload data set size), STG_SIZE (Staging data set size), and LOWOFFLOAD.

To minimize the amount of data offloaded from DFHLOG, you must:
1. Define a suitably-sized staging data set.
2. Ensure that the log-tail deletion process is working effectively.

CICS log manager controls the size of the system log stream by regularly deleting the oldest completed unit of work records. This operation is associated with activity keypoints. It is important, therefore, that you choose the correct activity keypoint frequency (AKPFREQ), that is, one that allows CICS to keep the system log down to a reasonable
size. If a system log stream exceeds the primary storage space allocated, it spills onto the secondary storage. The resulting I/O activity can adversely affect system performance. If the interval between activity keypoints is long, the volume of data could affect restart times. In general, an activity keypoint interval should be longer than the elapsed time of most transactions. The log tail is the oldest end of the log. At each activity keypoint, the CICS recovery manager requests the log manager to delete the tail of the system log.

Avoid “staging-data-set-full” events. A staging-data-set-full event occurs when a log stream’s staging data set becomes full before offloading of the data has completed.

Offloading is the movement of log data from the primary storage to the offload data sets. The size of the offload data sets needs to be reviewed. The offload data sets should be large enough to avoid too many DASD shifts.

For the CICS system log, the best performance is achieved when CICS can delete log-tail data that is no longer needed before it is written to secondary storage by the MVS System Logger. To monitor that this is being achieved, we need to examine the numbers in the CICS PA fields Count with DASD Write and Count without DASD Write.

These deletion values indicate:

- **Count without DASD Write**: Data was deleted from primary storage without first being written to DASD offload data sets. For a system log stream, this value should be high in relation to the value of “No. with DASD Write”.

- **Count with DASD Write**: Data was deleted from primary storage after being written to DASD offload data sets. For a system log stream, this value should be low in relation to the value in “No. without DASD Write”.

### 9.4 Scenario run

We started by defining the DASD-only log streams for DFHLOG. To do this, we used the MVS utility IXCMIA.PU. The name we chose, CICSTS.SCSCPAA1.DFHLOG, matched the name we defined to the CICS system through the stream name in the JOURNALMODEL definition (see Figure 9-2).

```plaintext
OVERTYPE TO MODIFY CICS RELEASE = 0530
CEDA Alter Journalmodel( DFHLOG )
Journalmodel : DFHLOG
Group : LOGTEST
Description ==> Journalname ==> DFHLOG
Type ==> Mvs Mvs | Smf | Dummy
Streamname ==> CICSTS.SCSCPAA1.DFHLOG
SYSID=PAA1 APPLID=SCSCPAA1
PF 1 HELP 2 COM 3 END 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

*Figure 9-2  JOURNALMODEL definition*

Example 9-1 shows the initial definition used for the log stream. This definition uses a number of default values.
Example 9-1  Definition of a DASD-only log stream

```cics_performance_analyzer
//MSLDEFIN EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=* , DCB=RECFM=FBA
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(CICSTS.SCSCPAA1.DFHLOG)
   DASDONLY(YES)
   MAXBUFSIZE(64000) STG_SIZE(3000)
   LS_SIZE(50)
   OFFLOADRECALL(NO)
   LOWOFFLOAD(20) HIGHOFFLOAD(80)
/*
For this exercise, we used values for the logger options as recommended by the CICS TS manuals. In the case of LOWOFFLOAD and STG_SIZE, the initial calculation is provided after running the DFHLSCU utility or using the formulae available in the CICS manuals. The LS_SIZE should be large enough so that each data set can contain multiple offloads of the primary storage.

The execution of the job in Example 9-1 resulted in the logstream definition shown in Example 9-2, which shows all the parameters and their values.

Example 9-2  Log stream defined

LOGSTREAM NAME(CICSTS.SCSCPAA1.DFHLOG) STRUCTNAME() LS_DATACLAS()
   LS_MGMTCLAS() LS_STORCLAS() HLQ(IXGLOG) MODEL(NO) LS_SIZE(50)
   STG_MGMTCLAS() STG_STORCLAS() STG_DATACLAS() STG_SIZE(3000)
   LOWOFFLOAD(20) HIGHOFFLOAD(80) STG_DUPLEX(YES) DULEXMODE(UNCOND)
   RMNAME() DESCRIPTION() RETPD(0) AUTODELETE(NO) OFFLOADRECALL(NO)
   DASDONLY(YES) DIAG(NO) LOGGERDUPLEX() EHLQ(NO_EHLQ)
   MAXBUFSIZE(64000)
/*
For the first run, we used the SIT parameter values of:
   ▶  AKPFREQ = 4000
   ▶  LGDFINT = 30
These are the default values for CICS TS V1.3.

We used TPNS to run the VSAM application to create a workload on the CICS region and then used CICS PA and its Logger reports to monitor the logger SMF records. For each test run, we ran the workload for 20 minutes. Each report was run for a 15 minute period taken from the middle of the 20 minute run.

To work with the System Logger records, we added the Logger System to CICS PA (Figure 9-3). We specified `new sc66logr` to add the Logger System to the CICS PA System Definitions. Here we added the Logger System manually. We could have used the Take-up facility as we did in other chapters in this book.
```
Figure 9-3  System Definitions screen

Figure 9-4 shows that we added the SMF data set containing the Logger SMF records.

Figure 9-4  Adding the SMF data set

Here we used SMFDATA.ALLRECS.G8525V00 for the SMF records.

Next, we needed to define a CICS PA Report Set so that we could report on the Logger usage. This was done in the CICS PA Report Set screen (Figure 9-5), where we defined a new Report Set called logreps.

Figure 9-5  New Report Set
We then changed the Global Options screen to add the Logger system about which we wished to run reports. Figure 9-6 shows the Global Options screen.

![File Systems Options Help](image)

**LOGREPS - Global Options**

**System Selection:**
- CICS APPLID
- DB2 SSID
- MQ SSID
- Logger

**Report Formatting Options:**
- Print Lines per Page: 60 (1-255)
- Time Zone
- Date Delimiter: /
- Time Delimiter: :

![Figure 9-6 Global Options screen](image)

We then went into the System Logger part of the Report Set so that we could configure it as shown in Figure 9-7.

![File Systems Confirm Options Help](image)

**CICS PA Report Set**

**Enter "/" to select action.**

- **Options**
- **Selection Criteria**
- **Performance Reports**
- **Exception Reports**
- **Transaction Resource Usage Reports**
- **Subsystem Reports**
- **DB2**
- **WebSphere MQ**
- **System Reports**
- **System Logger**
- **Performance Graphs**
- **Extracts**

**End of Reports**

![Figure 9-7 System Logger](image)

We then requested the Summary Report for the log stream CICSTS.SCSCPAA1.DFHLOG as shown in Figure 9-8.
We then entered line action RUN on the Report Set as shown in Figure 9-9. When pressing Enter, another screen opened which gave the chance to change some of the options, such as to Logger System or the data and time for selection. After we created the Report Set to produce the data in the form we required, on any subsequent test when the SMF data is located in a different SMF data set, we added the SMF data set to the system definition on the System Definition screen.

Example 9-3 shows the CICS PA parameters generated by the options we set. Note that we are selecting the log stream that is of interest to us.
CICSPA IN(SMFIN001),
NOAPPLID,
LINECNT(60),
FORMAT(';', '/', ),
LOGGER(OUTPUT(LOGR0001),
EXTERNAL(CPAXW001),
SUMMARY,
SORT(LOGSTREAM),
LOGSTREAM('CICSTS.SCSCPAA1.DFHLOG'),
TITLE1('CICS PA System Logger Report '))

Figure 9-10 shows part of the listing that was produced.

<table>
<thead>
<tr>
<th>V1R3MD</th>
<th>CICS Performance Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>System Logger - Logstream Summary</td>
</tr>
</tbody>
</table>

LOGR0001 Printed at 18:06:24 10/21/2003
Data from 17:45:00:20 10/21/2003 to 17:55:00:01 10/21/2003
CICS PA System Logger Report

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>MVSID</th>
<th>Structure name</th>
<th>First interval start</th>
<th>Last interval stop</th>
<th>Total Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS.SCSCPAA1.DFHLOG</td>
<td>SC66</td>
<td><em>DASDONLY</em></td>
<td>17:40:00.00 10/21/2003</td>
<td>17:55:00.00 10/21/2003</td>
<td>0000:15:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IXGWRITES</th>
<th>Bytes Writn to Total Count</th>
<th>Average Bytes</th>
<th>Count Write with DASD</th>
<th>Count Write w/o DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>22980</td>
<td>12669K</td>
<td>551</td>
<td>94183K</td>
</tr>
<tr>
<td>Rate(Sec)</td>
<td>25</td>
<td>14077</td>
<td>104648</td>
<td>14</td>
</tr>
<tr>
<td>Minimum</td>
<td>7517</td>
<td>4067508</td>
<td>30798K</td>
<td>2586</td>
</tr>
<tr>
<td>Maximum</td>
<td>7758</td>
<td>4325606</td>
<td>31797K</td>
<td>4107</td>
</tr>
</tbody>
</table>

| EVENTS | Demand Staging Offloads Threshld Demand Staging Offloads Demand Init'd |
|--------|-----------------------------|-----------------------------|-----------------------------|
| Offloads | 11                          | 1458                        | 0                           | 0           |
| Rate(Sec)| 0                           | 1                           | 0                           | 0           |
| Minimum | 3                            | 0                           | 7                           | 0           |
| Maximum | 4                            | 0                           | 17                          | 0           |

<table>
<thead>
<tr>
<th>EVENTS</th>
<th>Demand DASD Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type1</td>
<td>Type2</td>
</tr>
<tr>
<td>Total</td>
<td>48</td>
</tr>
<tr>
<td>Rate(Sec)</td>
<td>0</td>
</tr>
<tr>
<td>Minimum</td>
<td>9</td>
</tr>
<tr>
<td>Maximum</td>
<td>21</td>
</tr>
</tbody>
</table>

The fields that we concentrated on in the listing were, under DELETIONS:

- **Count With DASD Write**: The number of deletes from interim storage written to DASD
- **Count Without DASD Write**: The number of deletes from interim storage without having been written to the log data set

Under EVENTS, the fields were:

- **Offloads**: The number of times the log stream was offloaded.
- **Staging Thresholds**: The number of times the System Logger detected a Staging Data Set Threshold Hit condition (HIGHOFFLOAD reached) for the staging data set.
Demand DASD Shifts: The number of logstream DASD shifts (additional log data set allocates) initiated by this system. For DFHLOG and DFHSHUNT, this value should be small. Otherwise too much data is being offloaded. (You must check the LS_SIZE parameter for the logstream definition.)

Under DASD Writes, the field is:

Waits: The number of times the System Logger had to suspend processing before writing to DASD because a previous DASD write request has not completed.

We had 39 DASD shifts. Frequent DASD shifts have a negative effect on performance and expose the system to a depletion of the offload data sets. The number of offload data sets is limited by the logger DSEXTENT value. In an effort to decrease the number of DASD shifts, we redefined the DFHLOG, now with an LS_SIZE of 500. Increasing the LS_SIZE and so the size of the logstream DASD data set should reduce the number of DASD shifts as the logstream data set should now be able to contain more data before it needs to be offloaded. Example 9-4 shows the new logstream definition.

Example 9-4   Specifying LS_SIZE

//MSLDEFIN EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*,DCB=RECFM=FBA
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(CICSTS.SCSCPAA1.DFHLOG)
   DASDONLY(YES)
   MAXBUFSIZE(64000) STG_SIZE(3000)
   OFFLOADRECALL(NO)
   LS_SIZE(500)
   LOWOFFLOAD(20) HIGHOFFLOAD(80)
Figure 9-11 shows the results from the next run of the workload. This change improved the values for Offloads, Staging Threshold and DASD shifts. We then concentrated on just the Staging Threshold. The Staging Threshold indicates the number of times we hit the HIGHOFFLOAD value. In an attempt to reduce this number, we changed the SIT value AKPFREQ to 200. A lower AKPFREQ should mean that we carry out log-tail deletion more often and as such do not hit the HIGHOFFLOAD value as often.

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>MVSID</th>
<th>Structure name</th>
<th>First interval start</th>
<th>Last interval stop</th>
<th>Total Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS.SCSCPAA1.DFHLOG</td>
<td>SC66</td>
<td><em>DASDONLY</em></td>
<td>18:20:00.00 10/21/2003</td>
<td>18:35:00.00 10/21/2003</td>
<td>0000:15:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Count</th>
<th>Total Bytes</th>
<th>Average Bytes</th>
<th>Bytes Write to Interim Storage</th>
<th>Count</th>
<th>Count</th>
<th>Bytes Write</th>
<th>Bytes Write w/o DASD</th>
<th>Bytes Write w/ DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>18589</td>
<td>10425K</td>
<td>561</td>
<td>76198K</td>
<td>10452</td>
<td>6025</td>
<td>42856K</td>
<td>24682K</td>
</tr>
<tr>
<td>Rate/Sec</td>
<td>20</td>
<td>11583</td>
<td></td>
<td>84664</td>
<td>11</td>
<td>6</td>
<td>47610</td>
<td>27435</td>
</tr>
<tr>
<td>Minimum</td>
<td>2312</td>
<td>1241038</td>
<td></td>
<td>946952</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Maximum</td>
<td>8141</td>
<td>4607370</td>
<td></td>
<td>33734K</td>
<td>7027</td>
<td>3896</td>
<td>28811K</td>
<td>15962K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Count</th>
<th>Total Bytes</th>
<th>Average Bytes</th>
<th>Bytes After DASD Write</th>
<th>Count</th>
<th>Count</th>
<th>Bytes After DASD Write w/o DASD</th>
<th>Bytes After DASD Write w/ DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate/Sec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9-11 System Logger Logstream Summary report**
Figure 9-12 shows the result of the change. This change reduced the number of Staging Threshold hits and also reduced the number of offloads and the number of DASD shifts (reduced it to 0). The largest change it made was to the number of deletions after DASD writes; this also has been reduced to 0. We then redefined the DFHLOG with a larger STG_SIZE of 9000 to try and reduce the number of offloads still further. The STG_SIZE specifies how large to make the staging data set. Making the staging data set larger decreases the number of times an offload needs to happen.
We ran the test transaction again and then viewed the CICS PA System Logger Summary report shown in Figure 9-13. Here you see that we reduced the number of offloads and the staging threshold.

We redefined the system log with a LOWOFFLOAD value of 40%. The test we ran showed that we had two offloads in a 15 minute period. IBM recommends that you try to limit the number of offloads to not more than one an hour.

DFHLSCU recommends a value of 40% for the LOWOFFLOAD value for DFHLOG. However in practice, we have seen that a value between 40% and 60% is a good value. This needs to be reviewed, since too low an offload value may result in physical offloading of log data from primary to secondary storage, after the System Logger offload process completes the physical deletion of any unwanted log data during offload processing. However, a value that is too high may mean that subsequent offload processing occurs more frequently, since less space is freed up from primary storage during an offload operation.

The LOWOFFLOAD value should be greater than the space required for the sum of: the system log data generated during one complete activity keypoint interval plus the system log data generated (between syncpoints) by the longest-running transaction.
Figure 9-14 shows the results. This change caused a slight improvement in the number of offloads.

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>MVSID</th>
<th>Structure name</th>
<th>First interval start</th>
<th>Last interval stop</th>
<th>Total Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS.SCSCPAA1.DFHLOG</td>
<td>SC66</td>
<td><em>DASDONLY</em></td>
<td>11:50:00.00 10/22/2003</td>
<td>12:05:00.00 10/22/2003</td>
<td>0000:15:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IXGWrites</th>
<th>DELETIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Total</td>
</tr>
<tr>
<td>Write</td>
<td>With</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Total</td>
<td>19803</td>
</tr>
<tr>
<td>Rate(/Sec)</td>
<td>22</td>
</tr>
<tr>
<td>Minimum</td>
<td>4032</td>
</tr>
<tr>
<td>Maximum</td>
<td>7960</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offloads</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Rate(/Sec)</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD Writes</td>
</tr>
<tr>
<td>Type1</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Rate(/Sec)</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>
Figure 9-15 through Figure 9-17 show the same information but in the List version of the CICS PA report.

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>Structure name</th>
<th>MVSID</th>
<th>Flag</th>
<th>Interval expired at</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS.SCSPPA1.DFHLOG</td>
<td><em>DASDONLY</em></td>
<td>SC66</td>
<td>Staging</td>
<td>11:55:00.00 10/22/2003</td>
<td>SP7.0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bytes Write to</th>
<th>Count With</th>
<th>Count Without</th>
<th>Bytes Write to Interm</th>
<th>Bytes Write to DASD</th>
<th>Bytes Offload w/o DASD</th>
<th>Bytes Offload w/ DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>4032</td>
<td>2376964</td>
<td>590</td>
<td>16609K</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logstream name</th>
<th>Structure name</th>
<th>MVSID</th>
<th>Flag</th>
<th>Interval expired at</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS.SCSPPA1.DFHLOG</td>
<td><em>DASDONLY</em></td>
<td>SC66</td>
<td>Staging</td>
<td>12:00:00.00 10/22/2003</td>
<td>SP7.0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bytes Write to</th>
<th>Count With</th>
<th>Count Without</th>
<th>Bytes Write to Interm</th>
<th>Bytes Write to DASD</th>
<th>Bytes Offload w/o DASD</th>
<th>Bytes Offload w/ DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>7811</td>
<td>5654754</td>
<td>724</td>
<td>32940K</td>
<td>0</td>
<td>6835</td>
<td>28467K</td>
</tr>
</tbody>
</table>

Figure 9-15  System Logger List report (Part 1 of 3)

Figure 9-16  System Logger List report (Part 2 of 3)
## 9.5 Conclusion

To summarize, we showed that the CICS PA System Logger reports enabled us to identify possible areas of improvement. Using the information that the reports provided, we tuned the CICS System Logger log stream to perform more efficiently.
Scenarios with CICS Transaction Gateway

This chapter investigates the performance of an application that is composed of a front-end part running in WebSphere Application Server on a Windows 2000 platform and a back-end part running in CICS Transaction Server (TS). The two components of the application communicate through IBM CICS Transaction Gateway (CTG). CTG can be run on either the z/OS or Windows 2000 platform. With CICS Performance Analyzer (PA) reports, we discover that the performance of this application can potentially be improved.

Note: The scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer reports. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, these scenarios and the results provided are for demonstration purposes only. They do not provide definitive results for a customer environment.
10.1 What is CICS Transaction Gateway

IBM CICS Transaction Gateway provides secure, easy access from Web browsers and network computers to CICS applications, using standard Internet protocols in a range of configurations. It is a robust and scalable complement to a Web server. You can implement it as an e-business connector for IBM WebSphere Application Server, which is a Java 2 Platform, Enterprise Edition (J2EE)-compliant run-time environment for Java servlets and Java enterprise beans.

To communicate with CICS, CTG provides external access interfaces. The external access interfaces allow non-CICS applications to access and update CICS resources by initiating CICS transactions or by calling CICS programs. When used in conjunction with CICS communication facilities, they enable non-CICS programs to access and update resources on any CICS system.

The CTG supports such activities as developing graphical user interface (GUI) front ends for CICS applications. It also allows integration between CICS and non-CICS systems.

The latest release of the CTG is V5.0.1, and the currently supported platforms are: z/OS, OS/390, Linux® for S/390, AIX®, HP-UX, Sun Solaris, Windows NT®, and Windows 2000, and XP. CTG is supported for use with CICS/ESA V4.1, CICS/VSE 2.3 and CICS TS for VSE/ESA V1, but only if the CICS Transaction Gateway runs on a distributed platform. For use with CICS TS V1.2 for OS/390 or CICS TS V2 for z/OS, the CTG can run on z/OS, OS/390, or a distributed platform.

10.1.1 Gateway components and downstream protocols

CTG consists of the following principal components:

► **Gateway daemon**: This daemon listens on a Transmission Control Protocol/Internet Protocol (TCP/IP) port waiting for incoming requests from Java client applications.

► **Java class library**: This default /usr/lpp/ctg500/ctg/classes directory contains the following JAR files:
  - `ctgclient.jar`: Java class library
  - `ctgserver.jar`: Classes used by the Gateway daemon and for local Gateway support
  - `ctgsamples.jar`: Samples
  - `ctgadmin.jar`: Trace admin client
  - `cicsj2ee.jar`, `ccf2.jar`, `connector.jar`: J2EE classes

► **Client daemon**: This daemon provides client/CICS server connectivity.

Figure 10-1 shows the main components of the CICS Transaction Gateway.

![Figure 10-1 Components of the CICS Transaction Gateway](image-url)
Notice that the client daemon is not used on CTG for z/OS and that a direct access to a 3270 transaction is only possible through a CTG on a distributed platform (non-390).

The Gateway daemon is a long-running process. It functions as a server to network-attached Java client applications (such as applets or remote applications) by listening on a specified TCP/IP port. CTG supports downstream four different CTG network protocols (TCP, Secure Sockets Layer (SSL), Hypertext Transfer Protocol (HTTP), and Secured HTTP (HTTPS)).

You do not have to start the Gateway daemon when a Java client application executes on the same machine where the CTG is running. In this situation, you can use the CTG local protocol, which directly invokes the underlying transport mechanism using the Java Native Interface (JNI) module CTGJNI.dll.

### 10.1.2 Application programming interfaces and upstream protocols

The CTG provides three application programming interfaces (APIs) to client applications:

- **External Call Interface (ECI)** is a call interface to COMMAREA-based CICS applications. On z/OS and OS/390, ECI calls are mapped to External CICS Interface (EXCI) calls that provide similar functionality.

- **External Presentation Interface (EPI)** provides an API to invoke 3270-based transactions (CTG on distributed platform only).

- **External Security Interface (ESI)** is an API that allows password expiration management (PEM) functions to be invoked in CICS, to verify and change user IDs and passwords (CTG on distributed platform only).

On a distributed platform, the CTG client daemon provides upstream connectivity using the following network protocols:

- APPC connections from Windows and AIX platforms to all CICS platforms

- TCP62 (LU 6.2 over IP) connections to CICS/ESA V4.1, CICS TS V1.2 and CICS TS V1.3 for OS/390, and CICS TS for z/OS V2

- TCP/IP connections to CICS TS for z/OS V2.2, CICS TS for VSE/ESA V1.1.1, the TXSeries CICS Servers (AIX, Sun Solaris, Windows NT, Windows 2000, and HP-UX) and CICS OS/2 Transaction Server

Figure 10-2 shows a variety of ways to connect WebSphere Application Server to CICS TS for z/OS or CICS TS for OS/390.
10.2 Scenario description

To provide performance data for a CICS Transaction Gateway scenario, we created the environment shown in Figure 10-3.

The front-end application running in WebSphere Application Server on Windows 2000 can connect to the back-end CICS application program TRADERBL using either a CTG V5 for z/OS or CTG V5 for Windows 2000.
We did not attempt to workload manage the ECI or EXCI requests. We connected the CTG for z/OS to a target CICS TS V2.2 system PJA6 and we connected the CTG for Windows 2000 to a target system PJA7.

The application we used in this scenario is fully described in Chapter 10 of Enterprise JavaBeans for z/OS and OS/390 CICS Transaction Server V2.2, SG24-6284.

10.2.1 Front-end application

We used a session enterprise bean, named TraderAgent, to link to the back-end application TRADERBL. The enterprise bean is using the ECIRequest Java class to use the ECI interface, which is an interface to COMMAREA-based CICS programs. To direct the ECI requests to the distributed CTG and to the CTG that is running on the z/OS platform, we modified the environment entries of the enterprise bean using the WebSphere Application Server Application Assembly Tool. When the enterprise bean was developed, the environment entries were defined externally. Therefore we can change them without recompiling the bean itself.

We used the following environment entries for this version of the session bean:

- The name of the back-end program
- The CICS SYSID of the target CICS region
- The name of the mirror transaction
- The CTG URL

We changed the name of the CTG URL to direct ECI requests to our distributed CTG or to CTG for z/OS. You can call the enterprise bean by using two different enterprise bean clients:

- We have a Web application available that is using a servlet that is acting as an enterprise bean client for the enterprise bean that we are using. The Web application provides an Hypertext Markup Language (HTML) page that allows us to click the name of the ECIRequest application. When the ECIRequest application is selected, a servlet is started, which in turn, invokes the TraderAgent enterprise bean.

- To provide batch workload for the CTG performance scenario, we used a Windows enterprise bean client to execute the TraderAgent enterprise bean. We arranged a batch command file that allows to specify a loop-counter and a user ID to run the client several times.

10.2.2 Back-end application

We used the Trader application as a target back-end application for the CTG performance scenario. Trader is a sample share trading application that was used previously in other Redbooks projects. Trader, written in COBOL, uses the VSAM access method for file access and the CICS 3270 BMS programming interface.

The application consists of two modules: TRADERPL, which contains the 3270 presentation logic; and TRADERBL, which contains the business logic. TRADERPL invokes TRADERBL using an EXEC CICS LINK and passing a COMMAREA structure for input and output. TRADERBL contains logic to query and write to the persistent VSAM data, stored in two files: the company file and the customer file. We do not use TRADERPL, the presentation logic, for our CTG performance scenario. Instead, we use the TraderAgent enterprise bean. It passes the COMMAREA along with the ECI request and replaces the 3270 presentation logic.
The following business functions are provided by the COBOL application:

- Get_Company: Query the list of companies
- Share_value: Retrieve current stock quote from file
- Buy_Sell: Trade shares in a given company

We indicate which function we want to call by specifying the corresponding value in the COMMAREA.

### 10.2.3 CTG for z/OS EXCI scenario

We use two different scenarios for the CTG performance measurement. When we direct the ECI request to the CTG on z/OS, the ECI calls used by the application are mapped to the External CICS Interface. If we use the distributed CTG, we use the ECI over TCP/IP function to call the Trader application.

The EXCI interface is analogous to ECI. It allows programs that are running on z/OS, for example, batch programs or the CTG, to call CICS programs. The programs can transfer data using a COMMAREA. The EXCI allows a user to allocate and open sessions (or pipes) to a CICS region, and to pass distributed program link (DPL) requests over them. The multiregion operation (MRO) facility of CICS inter-region communication (IRC) facility supports these requests, and each pipe maps onto one MRO session, with a limit of 100 pipes per EXCI address space.

### 10.2.4 Distributed CTG using ECI over TCP/IP

When we direct the ECI request to the distributed CTG, we use the ECI over TCP/IP function to call the Trader application TRADERBL. ECI over TCP/IP allows direct access to CICS applications over TCP/IP. It removes the necessity to either use SNA or to configure TCP62 and the AnyNet® feature of VTAM.

CICS TS releases earlier than CICS TS for z/OS V2.2 do not support ECI over TCP/IP.

### 10.3 Scenario run

To produce SMF performance records for the CTG scenario, we used a batch CMD file, tsbc_ECI.cmd, that calls the Trader client from any Windows command prompt. We modified the environment variable JAVA_J2EE to point to our Java2 Enterprise Edition classes. Then we checked that the path specified in the CLIENTCLASSPATH environment variable contains the TraderEciClient.jar file. Example 10-1 shows the CMD file that we used to start the TraderEci client. The client is started using the Java command and the -classpath parameter.

```
Example 10-1 Batch CMD-file to run TraderClient
@echo off
rem """"""--""
rem TraderEci run command script
rem Use this file to call TraderClient from any DOS command line.
rem """"""--""
rem Modify the following to match your directory containing j2ee.jar:
set JAVA_J2EE="c:\j2sdkeel1.3.1\lib"
rem """"""--""
```
set CLIENTCLASSPATH=.;TraderEciClient.jar;TraderEciEJB.jar;%JAVA_J2EE%\j2ee.jar;..\ExternalJ ARs\CICSEJBClient.jar
rem ---------------------------------------------------------------
java -classpath %CLIENTCLASSPATH% itso.cics.cts22.trader.test.TraderClient %1 %2
rem ---------------------------------------------------------------

The Java command requires two parameters:

- The name of the properties file (specifying the JNDI parameters)
- The name of the user who is going to trade shares (user ID)

The sample TraderClient.properties file is configured to work with the TraderEci session bean deployed to WebSphere Application Server on our Windows 2000 Server.

We wanted to start the TraderECI client a number of times. Therefore, we created another CMD file, WLM_ECI.CMD, so we could start the CMD file shown in Example 10-1 automatically rather than manually.

The CMD file shown in Example 10-2 uses two parameters:

- A loop count in which you can specify the number of times to call the TraderECI client
- The prefix of the user ID you are going to use

The name of the user ID is built by the user ID prefix plus the loop count. Therefore, if you specify parameters 3 and MYUSER, then the following user IDs are used: MYUSER1, MYUSER2, and MYUSER3.

Example 10-2   Batch workload CMD file for TraderECI client

@echo off
if "%1" == "" goto :out
if "%2" == "" goto :out
CD C:\itsocts22\ejb-components\ECI
for /L %%f in (1,1,%1) do call tsbc_eci.cmd TraderClient.properties %2%%f
@echo tsbc.cmd*: Finished.
@end
@echo.
@echo parameters required!
@echo.
@echo use ECI_WLM loopcount userid-prefix
@echo.
@echo for example ECI_WLM 10 myid (loops ten times using myid1...myid10)
@echo.
@echo remember that ECI_WLM 100000 MYUSER would exceed the 8 byte userid length.
@echo.
@echo ECI_WLM 100000 ID will be ok...
@echo.
@end
CD C:\itsocts22\WLM_BATCH_CMDS

10.3.1 Running the workload using the CTG for z/OS

We used the WLM_ECI.CMD file to run a batch of 50 TraderEciClient applications. Each client application issues eight ECI request calls to back-end application TRADERBL. Each
ECI request call is received by the long running gateway daemon of the CTG for z/OS. The CTG for z/OS then uses the EXCI interface to call the CICS back-end application.

Example 10-3 shows the messages that we received during the process of one TraderEciClient run.

Example 10-3  WLM_ECI.CMD output

```
C:\itsocts22\WLM_Batch_CMds>eci_wlm 50 userid
*tsbc.cmd*: Starting the EJB client program.
Now starting a session with our TraderAgent using the user 'USERID1'.
EJBHelper::jndiLookup: Looking up home interface with JNDI name: domain/legac
et/itso/cics/cts22/trader/eci/DistributedTraderAgentHome
EJBHelper::jndiLookup: Creating an InitialContext of type: com.sun.jndi.cosna
g.CNctxFactory
Querying the version of our TraderAgent:

package itso.cics.cts22.trader.eci

Asking our TraderAgent for a list of all known companies:
Casey_Import_Export
Glass_and_Luget_Plc
Headworth_Electrical
IBM

Selecting the last company as our current company...
The last company is named IBM, here are its details:
The price per share in dollars = $163.0
The total of outstanding shares = 123456789 shares
The total outstanding share value = $2.0123456607E10
The price one day ago = $163.0
The price two days ago = $162.0
The price three days ago = $160.0
The price four days ago = $161.0
The price five days ago = $159.0
The price six days ago = $156.0
The price seven days ago = $157.0
The cost to sell a share = $400.0
The cost to buy a share = $0.0

The portfolio of USERID1 contains 28 shares in IBM worth $4564.0.
Selling 23 shares...
Buying 16 shares...
The portfolio of USERID1 now contains 35 shares in IBM worth $5705.0.
```

When the batch of 50 TraderEciClient applications completed, we switched the current SMF data set using the /I SMF command. Then we used the Take-Up from SMF function to update our CICS Performance Analyzer system definitions. We performed the following steps to do the Take-up from SMF:
1. On the Primary Option Menu, select option 1 (System definitions).
2. On the System Definition menu, select option 4 (Take-Up from SMF).
3. Specify the SMF File for data take-up and press Enter. A job was generated that processes Take-Up from SMF.
4. Return to the entry of system definitions. It displays a screen that tells you that take-up data is not yet merged to system entries. Press Enter to update the system entries.
5. On the System Definition menu, select option 2 (Maintain SMF files) to check that the list of SMF file is updated with the current SMF data set.
6. Return to the System Definition menu and select option 1 (Define Systems, SMF Files and Groups). We selected CICS system SCSCPJA6 from the list and added the current SMF data set.

Creating a Performance Summary report

To call the back-end program using the EXCI interface, CICS attaches a mirror transaction, which is CSMI by default or your own mirror transaction. Therefore, we are interested in response time and CPU time of mirror transaction CSMI. The first report we created was a Performance Summary report. We performed the following steps to create a Performance Summary report:

1. On the primary options menu, select option 2 (Report Sets).
2. Type NEW in the command line to create a new Report Set.
3. On the next screen, type the name of the Report Set CTGZOS. Press Enter.
5. On the Performance Summary Report screen, specify the APPLID of our CICS system SCSCPJA6. Then select the existing Report Form VSAMSUM.

Refer to Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139, to find a description of how we created the VSAMSUM Report Form. We typed line action $ next to the performance option in group selection criteria and specified a report interval and selection criteria for transaction CSMI as illustrated in Figure 10-4.

```
| Command ===>| CTGZOS - Performance Select Statement Row 1 of 3 More: > Scroll ===>| CSR
| Inc Field | --- Value or Range --- Object |
| Inc Name + Type Value/From To List + |
| INC TRAN CSMI |
```

Figure 10-4 Performance Select Statement screen
6. Return to the Report Set screen. Entered line action RUN on Summary typed SUB in the command line to run the Report Set.

When the job completed, it produced the output shown in Figure 10-5.

<table>
<thead>
<tr>
<th>Tran</th>
<th>Avg Response Time</th>
<th>Max Response Time</th>
<th>Avg Dispatch Time</th>
<th>User CPU Time</th>
<th>Avg Suspended Time</th>
<th>FC Total Time</th>
<th>RLS Wait Time</th>
<th>Avg FC Wait Time</th>
<th>Avg FCADD Time</th>
<th>Avg FCDELETE Time</th>
<th>Avg FCGET Time</th>
<th>Avg FCPUT Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMI</td>
<td>.0031</td>
<td>.1121</td>
<td>.0015</td>
<td>.0008</td>
<td>.0015</td>
<td>3</td>
<td>.0015</td>
<td>.0000</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 10-5  Performance Summary report for CSMI transaction

The Summary report gives a first impression about CICS internal response times and CPU consumption. For a closer look at the CICS internal performance of the CSMI transaction, we provide a Performance List report of CSMI transaction.

Creating a Performance List report
We performed the following steps to create a Performance List report:

1. On the primary options menu, select option 2 (Report Sets). When the list of available Report Sets is displayed, select Report Set CTGZOS.

2. Deactivate the summary report that you created earlier when you typed line action D next to the summary report option. The Yes in the Active column after the summary option is changed to No.

3. Type line action S next to the list option in group performance reports and press Enter.

4. On the performance list report screen, specify APPLID SCSCPJA6 and Report Form VSAML. Refer to Chapter 6, “VSAM application performance analysis and Transaction Resource Monitoring support” on page 139, which describes how to create Report Form VSAML. Report Form VSAML uses the same sequence of performance fields as VSAMSUM.

5. Still on the Performance List Report screen, type line action S next to the performance option in group selection criteria. We specified selection criteria for transaction CSMI and the report interval.

6. Return to the Report Set screen from where you ran the Report Set. A Performance List report job was generated and executed. Figure 10-6 shows the Performance List report that we created.
We were surprised that the list contained about 400 mirror tasks. We did not expect mirror
tasks to get detached since we specified DFHSIT parameters MROLRM (long running mirror)
and MROFSE. MROFSE specifies whether you want to extend the lifetime of the long-running
mirror to keep it allocated until the end of the task rather than after a user sync point for
function shipping applications.

We started a CEDX session on transaction CSMI to look more closely at the EXEC CICS
LINK requests that were issued over the EXCI interface.

Figure 10-7 shows the CEDX screen when it stopped at one of the EXEC CICS LINK
commands. The LINK command actually was originated when the TraderEciClient issued an
ECI request call to link to the back-end program TRADERBL. The fact that the LINK
command was issued with the SYNCONRETURN option implies that the TraderEciClient is
not running in extended mode. In traditional CICS client ECI applications, extended mode
encompasses a series of one or more ECI requests to a server program, each executed with
SYNCONRETURN set to off, followed by a final ECI request (to the same server program)
that is executed with SYNCONRETURN set to on. This final ECI call causes CICS to take a
sync point on successful completion of the server program, and any changes to resources
made by the server program to be committed.
We looked at the Java source of the enterprise bean methods that issue the ECI request call. We found that all of them specified the ECI_NON_EXTEND parameter as well as the ECI_LUW_NEW parameter. This in fact is the reason that each EXCI request runs with SYNCONRETURN set to on, which terminates the mirror transaction even if MROLRM SIT parameter is set to YES. Example 10-4 shows how the ECI request call has been coded. To avoid detaching the mirror transaction, change the recovery mode to ECI_EXTENDED mode. The last ECI request in series of ECI requests should issue a COMMIT request which results in a sync point.

Example 10-4  ECI request call issued by TraderAgent enterprise bean

```java
// create ECI request
request = new ECIRequest(
    ECIRequest.ECI_SYNC, // simple ECI synchronous call
    aCicsServer,        // servername as configured to CTG
    null,               // no userid
    null,               // no password
    aProgramName,       // name of the CICS program to run
    aTranId,            // mirror transaction ID
    commarea, commarea.length, // COMMAREA and length
    ECIRequest.ECI_NO_EXTEND, // one LUW per call
    ECIRequest.ECI_LUW_NEW);  // default LUW ID for new call
```

10.3.2 Running the workload using the distributed CTG

To run our workload using the distributed gateway, we changed the CTG-URL variable of the enterprise beans environment entries. We specified the TCP/IP address of the Windows 2000 Server and the port on which the gateway daemon is listening. Figure 10-8 shows the Application Assembly Tool screen where we modified the environment entries of the DistributedTraderAgent enterprise bean.

Figure 10-8 also shows the remaining environment entries that we used. We specified the CICS SYSID PJA6, the program name TRADERBL, and the mirror transaction ID CPMI.
When the DistributedTraderAgent bean was modified to point to the correct CTG, we started our workload again. We used the same CMD-file that we used for the CTG for the z/OS scenario which is processing a batch of 50 TraderClient calls.

![Application Assembly Tool](image)

**Figure 10-8 Application Assembly Tool**

While the workload was running, we verified that it ran correctly by checking the output of the CMD-file as illustrated in Example 10-3 on page 248. When 50 calls to the TraderEciClient completed, we switched the current SMF data set to archive the data.

To investigate the CICS internal performance of ECI over TCP/IP, and to see a comparison between EXCI and ECI over TCP/IP interfaces, we provided a similar Performance Summary report that we used to show the performance of the CTG for z/OS.

When we configured CICS to use ECI over TCP/IP, we created a TCPIPSERVICE resource definition. We specified the ECI protocol which automatically inserted the name of the listener transaction CIEP to the definition. From the CICS internal performance point of view, we wanted to look at two transactions: CPMI which is the mirror transaction that we specified in environment entries of the enterprise bean and CIEP which is the listener transaction for the ECI over TCP/IP function. Figure 10-9 shows a CEMT I TCPIPS display of the TCPIPSERVICE definition we use for ECI over TCP/IP.
Refer to “Creating a Performance Summary report” on page 249 for details about how to create a performance summary report for the distributed CTG scenario. We performed the following steps to create the Performance Summary report for transactions CIEP and CPMI:

1. Use the Take-Up for SMF function to update the CICS system entries with new performance data.
2. Copy the existing Report Set CTGZOS as CTGECI.
3. Modify the existing selection criteria as illustrated in Figure 10-10.
4. Run the Report Set.

Figure 10-9  ECI over TCP/IP TCPIPSERVICE definition

Figure 10-10  Performance Select Statement screen
When the Performance Summary report job completed, it produced the output shown in Figure 10-11. We did a CPMI to CSMI comparison and found that average response time for CPMI transaction was higher. In any case, the average CPU time of 0.0008 for both transaction is very much the same. The maximum response time of a single transaction might have influenced the average response time of CPMI transaction as well. The workload scenario using the distributed CTG ran one hour later than the CTG for z/OS scenario, therefore other activities on our system may have influenced CICS performance. For example, the ratio between average dispatch time and average CPU time is not good. It appears that the CICS address space was not continuously dispatched when the workload was running.

![Figure 10-11 Performance Summary report](image)

As we did for the CTG for z/OS scenario, we produced a Performance List report for the distributed CTG scenario as well. Refer to “Creating a Performance List report” on page 250 for details about how we created a Performance List report for the CTG for the z/OS scenario.

We performed the following steps to create a Performance List report:

3. Create a Performance select statement.
4. Run the Report Set.

We used the same enterprise bean for the distributed CTG performance scenario. Therefore, we also discovered that we do not detach the mirror transaction. The Performance List report (Figure 10-10) contained about 400 mirror transactions.

To use long running mirror transactions for this scenario, we must change the ECI request calls issued by the enterprise bean. We must change the parameter ECI_NON_EXTENDED to ECI_EXTENDED. We must also change the logic of the enterprise bean slightly, since a token must be passed along with the request and the last ECI request call must be COMMIT or BACKOUT. In addition, we must enable the Gateway daemon support of z/OS Resource Recovery Services (RRS) using the CTG_RRMNAME environment variable.
10.4 Conclusion

Running this scenario, we showed that by analyzing CICS PA reports, a system programmer can deduce how different APIs are used by the CTG client application. In our case, an application change can be suggested that affects a certain performance improvement by using extended ECI calls. Of course, the reason extended ECI calls were not used in the first place was to prevent tying up CICS recoverable resources for the full duration of an extended unit of work.
CICS Web Support and 3270 Bridge

This chapter presents a brief overview of the CICS Web Support and the 3270 Bridge interface. It shows how CICS PA can help you to tune the interface by deciding on an appropriate setting of the SOCKETCLOSE parameter on the TCPIPSERVICE definition. This value specifies if, and for how long, CICS should wait before closing the socket, after issuing a receive for incoming data on that socket. For more information about the SOCKETCLOSE parameter and the various setting options, refer to the *CICS Transaction Server for z/OS CICS Resource Definition Guide*, SC34-5990.

**Note:** This scenario was used to provide a situation that allows us to demonstrate the use of CICS PA reports. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, these scenarios and the results provided are for demonstration purposes only. They do not provide definitive results for a customer environment.
11.1 CICS Web Support overview

CICS Web Support enables Web browsers to communicate directly with mainframe CICS application programs without an intermediate gateway or a separate Web server. When an end user selects a CICS Web Support Uniform Resource Locator (URL) from a browser window, the request is sent to CICS over TCP/IP using the HTTP protocol. In first instance, the request is accepted by a CICS internal control task which attaches a task to analyze the request as presented in the URL. The name of this attached task is CWXN by default but can be changed. Under control of the CWXN transaction, the appropriate application to be run is determined and an alias transaction is attached to run the application. The default name of the alias transaction is CWBA.

You can use a special CICS-provided application program, DFHWBTTA, to invoke a 3270 transaction in a 3270 Bridge environment. Thus the CICS Web Support exploits the capabilities of the bridge service to allow existing 3270-based transactions to be driven from the Web.

Figure 11-1 shows a general overview of this process. A Web request is initially taken by the Socket Listener task CSOL, which attaches the CWXN transaction to analyze the request. The requested user application program is LINKed to under control of the subsequently attached alias transaction CWBA. If the user application program to be executed is the CICS provided program DFHWBTTA, then the requested 3270 transaction is processed under a separately attached transaction that runs in a bridge environment.

11.2 Scenario description

For the purposes of this section, we did not use a tool to provide heavy workload. It is sufficient to run a few Web requests to show the possibilities of reporting with CICS PA.

We used two CICS systems, one CICS TS V1.3 system, SCSCPA6, and one CICS TS V2.2 system, SCSCPJ6A6. Both ran on the same z/OS system so that they used the same IP address, 9.12.6.29. For SCSCPA6, we defined that it is listening on port 8080 as you can see in the TCPIPSERVICE definition shown in Figure 11-2. Note also that the SOCKETCLOSE time is 20 seconds, which means that the socket stays open for 20 seconds to wait for another incoming requests after some data was received.
For SCSCPJA6, we specified that it was listening on port 8081 and we had SOCKETCLOSE equal to zero. This means that the socket closes immediately, if no more data is available after the first receive. Figure 11-3 shows the TCPIPSERVICE definition for SCSCPJA6.
11.3 Scenario run

We did not introduce a specific workload. From two workstations, we ran some applications. In SCSCPJA6, we used the following URLs:


For running applications on SCSCPAA6, we used:


With this as the starting point, we show you the Transaction Group Report. When a request from a browser comes into the CICS system, the CWXN transaction that handles the request is assigned a transaction group ID. This transaction group ID is used to correlate the transactions that CICS executes for the same incoming work request. You can ask CICS PA to provide a list of transactions that executed in the same group. The Transaction Group option is available on the Report Set screen in the Reports category. We created a new Report Set, called WEB. On the Report Set panel, we selected the Transaction Group option and received the screen shown in Figure 11-4. On this screen, we elected to have only those groups that contain more than one record.

```
File Systems Options Help
--------------------------------------------------------------------------
WEB - Transaction Group Report
Command ===> System Selection:                       Report Output:
            APPLID . .           +                  DDname . . . . . . . . . TRGP0001
            Image . . SC66      +                  Print Lines per Page . .      (1-255)
            Group . .           +

Processing Options:
1 1. Groups of more than one record
2 2. Groups of a single record
3 3. All Groups

Report Format:
Title . . Grouping Web transactions

Selection Criteria:
Performance
```

*Figure 11-4  Transaction group processing options*
We returned to the Report Set panel and ran the Web report. The beginning of the report is shown in Figure 11-5.

![Figure 11-5 Default Transaction Group output](image)

We concentrate on the Origin column in this example. This column gives the origin type from the SMF Transaction flags field, a string of 64 bits used for signaling transaction definition and status information. The values we are interested in are SOCKET, WEB and BRIDGE. To create a report with only transactions with one of those three types of origin, we again selected the Transaction Group option and selected the Performance selection criteria where we entered the selection to have only those records with ORIGIN is equal to WEB, SOCKET or BRIDGE as shown in Figure 11-6.

![Figure 11-6 Transaction Group Selection criteria](image)

File Edit Object Lists Options Help

--- Performance Select Statement ---

--- Value or Range ---

--- Object ---
After we ran the report again, we received the report shown in Figure 11-7.

The report starts with the groups of APPLID SCSCPA6. We see that there are many transactions running in the same group. This is the result of having a socket close time of 20 seconds. If the client sends a Connection: Keep-alive HTTP header with a SOCKETCLOSE value of 20 seconds, the socket stays open and every request coming in within 20 seconds
after the last receive is handled by the same CWXN transaction. For the transaction group containing the transactions with task numbers 45 to 55, we see that a CEMT transaction was started in a bridge environment. CWXN is the first transaction in the group. The second transaction is the CWBA transaction with task number 46. This transaction links to program DFHWBTTA that then attaches the CEMT transaction with task number 47. Transactions 47 to 55 represent a conversation with the CEMT transaction. The last receive from the socket resulted in the CWBA transaction with number 55 that ended at 11:13:51.34. Twenty seconds later, at 11:14:22.37, the CWXN transaction ends because of the socket close time of 20 seconds.

For the SCSCPJA6 system, which has a SOCKETCLOSE value equal to zero, we see that the CWXN transaction is ending immediately, resulting in one CWXN and CWBA transaction per incoming request.

At the end of the report, we also receive a summary per origin type.

11.4 Tuning your CICS Web Support environment

This section discusses how you can tune your CICS Web Support environment for better performance.

11.4.1 Storage consumption by CWXN

Authorized program analysis report (APAR) PQ33097 for CICS TS 1.3 explains that the default SOCKETCLOSE(NO) on the TCP/IP SERVICE resource definition may lead to MAXTASK or SOS conditions. This APAR changes the default value for the HTTPNSL TCP/IP SERVICE definition in group DFHSSOT from NO to zero.

Let's look at the storage consumption by the CWXN transaction. As a starting point, we use the CICS PA provided sample Report Forms USTORLST and SSTORLST. In the Report Forms screen, you can copy these forms to your own Report Forms data set by putting your cursor under the Samples option and selecting them from the resulting pop-up window. In your private data set, you can change them according to your needs. We decided to add the APPLID. As an example, we show the modified USTORLST in Figure 11-8.
We ran two Performance List reports specifying these form names in the form field on the Performance List Report screen. For the Performance selection criteria, we chose to only have the report for the CWXN transaction. Figure 11-9 shows the result of the run with the USTORLST form.

The number of GETMAINs and the high water marks for allocated storage in region SCSCPJA6 for CWXN are almost constant and low. In region SCSCPAA6, these numbers are
greater and related to the number of alias transactions that are attached for this run of CWXN. Figure 11-10 shows the equivalent report for shared storage usage.

![Figure 11-10 CWXN shared storage report](image)

You can see a similar trend here. The number of GETMAINs and storage high water marks are directly related to the number of CWBA transactions and to the SOCKETCLOSE value. Before coming to a conclusion, we should look at the CPU consumption of the CWXN transaction. Again, we used a CICS PA provided sample form, this time CPULST. We modified the USERID field to make it APPLID. Figure 11-11 shows the modified form.

![Figure 11-11 Modified CPULST Report Form](image)
Figure 11-12 shows the result of running a Performance List report with this form.

This report also shows longer response times for the SCSCPAA6 region because the socket does not close immediately. The value of the response time is dependent on the activity on the socket. The socket stays open for listening as long as requests come in within 20 seconds after the last receive. For SCSCPJA6, the response is dependent on the handling of a single request. Looking at the total CPU consumption, we see that the CPU time is relatively low. Most CPU time is used on the QR TCB, although some is spent on the SO TCB to handle the socket request. This CPU time is reported in the Miscellaneous CPU time field.

11.5 Conclusion

Since CPU time is not directly involved, tuning your CICS Web Support environment means that you need to find an optimum balance. You need to find it between the number of CWXN transactions that you allow to exist concurrently and the storage they occupy. You also need to consider the overhead of starting a new CWXN transaction each time.
Chapter 12. Java applications in CICS

This chapter compares the performance of a sample CICS-provided Java application in an environment of resettable Java Virtual Machine (JVM) to its performance in an environment of a non-resettable JVM. It also demonstrates the performance advantages of using sharable application classpaths.

**Note:** The scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer (PA) reports. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore, you should see these scenarios and the results provided for demonstration purposes only. They do not provide definitive results for a customer environment.
12.1 CICS and Java

You can write Java application programs that use CICS services and execute under CICS control. CICS Transaction Server (TS) V2.2 supports three programming models. All three models are supported by the new Persistent Reusable JVM that executes under CICS control.

12.1.1 Java language programs in CICS

CICS TS provides a Java class library called JCICS that covers many functions of the traditional EXEC CICS programming interface. The class library is shipped in a dfjcics.jar file. You can download it to your workstation for use in an Integrated Development Environment (IDE), such as WebSphere Studio Application Developer (WSAD). JCICS allows you to access CICS resources such as VSAM files, CICS transient data, and temporary storage queues. It also allows you to invoke other CICS programs and transactions.

Although these CICS Java programs run in a JVM environment, you invoke them the same way as other CICS programs written in procedural languages. You can invoke them as initial programs of a transaction through a transaction definition, through an EXEC CICS LINK or EXEC CICS XCTL with or without a COMMAREA, as a program that executes an EXEC CICS START TRANSACTION request, through APPC, through transient data queue triggering mechanism, and by all other available traditional means.

CICS TS V1.3 introduced CICS support of the Java language and the JCICS class library, they were further enhanced in CICS TS 1.3 by authorized program analysis report (APAR) PQ34321 to provide SQLJ and JDBC access to DB2 relational database.

12.1.2 Stateless CORBA objects

Stateless CORBA objects are Java server applications that are invoked by a client application using the Internet Inter-ORB Protocol (IIOP). No state is maintained in object attributes between successful invocation of methods.

Inbound CORBA object communication was introduced in CICS TS V1.3. In CICS TS V2.2, CORBA objects can also make outbound IIOP calls. Therefore, they can behave as a client or as a server within the scope of an Object Transaction Service (OTS) distributed transaction.

Stateless CORBA objects can use JCICS API to interact with CICS.

12.1.3 Enterprise JavaBeans

Enterprise JavaBeans (EJBs) are non-visual server-side components of a distributed transactional application that conform to Sun Microsystem’s Enterprise JavaBean Specification. CICS has implemented support for Version 1.1 level of this specification by mapping the interfaces defined in the specification to underlying CICS services.

You can develop Enterprise JavaBeans that use the JCICS class library to access CICS resources or programs directly. However, these applications are not portable to a non-CICS EJB run-time environment.

The EJB specification defines two types of Enterprise JavaBeans:

- **Session beans**: Encapsulate a session between a client and a server component. There are two varieties of session beans:
– **Stateless session beans** behave in a manner similar to Stateless CORBA Objects. That is, state data is not maintained between method invocations. Stateless session bean objects can handle multiple requests from multiple clients so sessions can be pooled. An OTS transaction cannot span though method invocations.

– **Stateful session beans** support multiple consecutive method invocations originating from the same client. The state data is maintained between method invocations. The bean object exists for the duration of a single client/server session and an OTS transaction can span multiple method invocations.

CICS fully supports both types of session beans. It is important to understand that a method invocation for both types of session beans maps to one CICS transaction, so the system behavior in both cases conforms to the CICS pseudo-conversational programming model.

- **Entity beans**: Usually encapsulate access to relational data. Entity beans are not supported by CICS TS. Relational data that usually resides in a local DB2 database can be accessed through JDBC or SQLJ interfaces.

Refer to Chapter 13, “Enterprise JavaBeans in CICS” on page 285, for further discussion.

### 12.2 Scenario description

This scenario uses the CICS-provided sample transaction that accesses a temporary storage queue. This sample shows how to use the TSQ class. It consists of a single transaction, JTS1, which invokes a single Java class, TSQ.ClassOne. It also uses an auxiliary temporary storage queue. Refer to *CICS Transaction Server for z/OS Java Applications in CICS*, SC34-6000, which fully describes how to generate, set up and run this sample TSQ application.

Because the sample application uses only one single Java class, we found it necessary to download the application to a workstation and repackage it by adding one more class to the source of the program. We added the instantiation of the second class. This addition did not change anything in the execution logic of the sample program, but simply caused two classes to be loaded in the JVM instead of one. You can run the sample program DFJ$JTS1 by using the transaction code of JTS1. All definitions are provided by CICS in the DFH$JVM CSD group.

To run a Java program in a JVM environment, you need a PROGRAM definition that specifies this program to be a JVM program (JVM set to YES). You must also specify the name of the application class in the JVMCLASS parameter and point to a JVM properties file in the JVMPROFILE parameter. Figure 12-1 shows the relevant parameters from the DFH$JTS1 PROGRAM definition.
The JVMPROFILE parameter specifies the name of the profile that CICS is to use to provide the JVM characteristics. The profile is located as a member of a partitioned data set with DD-name DFHJVM. In this member, the three important parameters to check for this test are:

- **JVMPROPS**: Specifies the full path of the system properties file that resides in hierarchical file system (HFS) and that CICS is to use when creating a JVM. On our system, we have: JVMPROPS=/u/cicsts22/props/scscpj##/dfjvmpr.props.

- **CLASSPATH**: Specifies the path to user applications. On our system, we have: CLASSPATH=/u/cicsls/:/usr/lpp/cicsts/cicsts22/samples.

- **Xresettable**: Specifies whether the JVM is eligible to be reused again for execution of other suitable JVM programs. In our test, we run with Xresettable=NO and Xresettable=YES.

We specify only the ibm.jvm.shareable.application.class.path property for this test, from the HFS JVM.properties file. If you use only the CLASSPATH in the profile in the DFHJVM data set, classes are reloaded from their HFS location each time the JVM is reused. Defining classes in the ibm.jvm.shareable.application.class.path system property provides additional optimization by caching the classes in the JVM. On our system, we have:

```
ibm.jvm.shareable.application.class.path=/u/cicsls/work/:/u/cicsls/samples/dfjcics/
```

Finally, there is one CICS system initialization parameter to be mentioned. MAXJVMTCBS specifies the maximum number of open task control blocks (TCBs) that CICS can create for use by JVM programs. We decided to run with the default value of 5.

In this test, we plan to perform three runs. A first run is with the Xresettable parameter set to NO so that for every JVM program execution, a new JVM has to be initialized. The second run should show the difference when running with Xresettable set to YES so that the JVM is initialized once and reused afterwards. During this run, we do not specify the ibm.jvm.shareable.application.class.path property. In the last run, we use cached classes by specifying the ibm.jvm.shareable.application.class.path property.

We used Teleprocessing Network Simulator (TPNS) to generate the workload for CICS. Fifty terminals were simulated that always start one and the same transaction. When transaction output is received, immediately a new transaction run was requested. The intention was to run the workload for about 15 minutes and take a sample of five minutes from the middle of the run.
12.3 Measuring JVM performance

This section compares the transaction performance in resettable and non-resettable JVM environment.

12.3.1 Xresettable=NO

After the SMF data sets were switched, we ran the Take-up facility on the SMF file. In CICS Performance Analyzer Primary Option Menu, we chose 1 to go to the Systems Definitions screen. On the Systems Definitions screen, we chose option 4 for the Take-up function. On the resulting screen, we entered the SMF records data set name and submitted the batch job. Figure 12-2 shows output from the take-up job.

```
VIR3M0 17:02:50 10/14/2003 CICS Performance Analyzer Page 1
Take-up from SMF

CPA2012I  Processing started for SMF file SMFIN001
CPA2013I  SMF records for System SC66 start at 10/2003 10:49:44.74
CPA2015I  DB2 Accounting record found, DB2 SSID=D7Q2 Release=7.1
CPA2014I  CMF record for CICS system found, APPLID=SCSCPBAA1 Release=5.3.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPTA1 Release=5.3.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPTA2 Release=5.3.0
CPA2016I  MVS System Logger record found, System=SC66LOGR
CPA2014I  CMF record for CICS system found, APPLID=SCSCPJA6 Release=6.2.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCLSA5 Release=6.2.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPJA1 Release=6.2.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPJA7 Release=6.2.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPJA2 Release=6.2.0
CPA2014I  CMF record for CICS system found, APPLID=SCSCPAME Release=6.2.0
CPA2013I  Processing ended for SMF file SMFIN001 - 11 system(s) found
CPA2000I  Take-up processing has completed, RC=0
```

Figure 12-2 Take-up output

Figure 12-3 shows the result of the take-up job on the Systems Definition screen. Because of the mix of systems and the amount of data, we decided to create a new data set containing only the records we were interested in. We used CICS PA Record Selection to do this.

On the Main Options screen, we selected 2 to go to the Report Sets screen. On the command line, we entered NEW JVM to create a new Report Set. We selected Record Selection in the Report category.
On the Report Selection Extract screen, we entered the APPLID and Image name as found during the take-up. We also entered the data set name that had to contain the subset of the SMF records, CICSLS2.JVM.FIRST.RUN, and specified 1 for the disposition to create a new data set. Since we planned to copy only records for transaction JTS1, we entered S next to the Performance selection criteria. Figure 12-4 shows the screen with its input.

On the Performance Select Statement screen (Figure 12-5), we entered the field name TRAN and its value of JTS1. We also entered I to include this selection.
We submitted the batch job. Figure 12-6 shows the result.

From now on, we use the data set CICSLS2.JVM.FIRST.RUN to produce our reports. We returned to the System Definitions screen where we selected our system entry to change the name of the SMF data set to the new data set that contains our subset of records. We easily selected the new data set name from the list of SMF data sets since its name was added to the list of SMF data sets by the Record Selection function.

Before running the first report, we chose a Report Form. We chose to use the CICS PA provided sample forms that are related to JVM. To bring these to our forms data set, on the CICS PA Primary Option Menu, we chose option 3 to go to the Report Forms screen. On this screen, we moved the cursor to the action bar under the Samples option and pressed Enter. We selected option 1 to populate the Report Forms data set with the sample forms (see Figure 12-7).
We scrolled down the list and selected the JVMLST and JVMSUM sample forms. We pressed F3 to return to our private Report Forms screen where we saw these forms included in the list.

We first produced the LIST report. We returned back to the Primary Option Menu, where we selected 2 to go the Report Set screen. On the Report Set screen, we entered NEW JVML to create a new Report Set. On the EDIT Report Set screen, we entered S next to the List option in the category Performance Reports. On the Performance List Report screen (Figure 12-8), we entered the APPLID and image name as generated by the take-up, specified to use the JVMLST Report Form, and entered a report title.

---

**Sample Report Forms**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNDLST</td>
<td>LIST</td>
<td>Transaction Abend List</td>
</tr>
<tr>
<td>ABNDSUM</td>
<td>SUMMARY</td>
<td>Transaction Abend Summary</td>
</tr>
<tr>
<td>BADCPU</td>
<td>LISTX</td>
<td>Top 20 Worst CPU Times</td>
</tr>
<tr>
<td>BADFILE</td>
<td>LISTX</td>
<td>Top 20 Worst File Requests</td>
</tr>
<tr>
<td>BADRESP</td>
<td>LISTX</td>
<td>Top 20 Worst Response Times</td>
</tr>
<tr>
<td>BADRMI</td>
<td>LISTX</td>
<td>Top 20 Worst CICS RMI Times</td>
</tr>
<tr>
<td>BADRMIQ</td>
<td>LISTX</td>
<td>Top 20 Worst CICS RMI Requests</td>
</tr>
<tr>
<td>BADSUSP</td>
<td>LISTX</td>
<td>Top 20 Worst Suspend Times</td>
</tr>
<tr>
<td>BADTDQ</td>
<td>LISTX</td>
<td>Top 20 Worst Tdqueue Requests</td>
</tr>
<tr>
<td>BADTSQ</td>
<td>LISTX</td>
<td>Top 20 Worst Tsqueue Requests</td>
</tr>
<tr>
<td>BTSACLST</td>
<td>LIST</td>
<td>CICS BTS Activity - Overview</td>
</tr>
<tr>
<td>BSTRQLST</td>
<td>LIST</td>
<td>CICS BTS Request Activity</td>
</tr>
<tr>
<td>BSTRQSUM</td>
<td>SUMMARY</td>
<td>CICS BTS Request Activity</td>
</tr>
<tr>
<td>COMMWLST</td>
<td>LIST</td>
<td>Transaction Comms Wait Analysis</td>
</tr>
<tr>
<td>COMMWSUM</td>
<td>SUMMARY</td>
<td>Transaction Comms Wait Analysis</td>
</tr>
<tr>
<td>CPULEXTR</td>
<td>LIST</td>
<td>CPU Analysis and Extract</td>
</tr>
</tbody>
</table>

---

We scrolled down the list and selected the JVMLST and JVMSUM sample forms. We pressed F3 to return to our private Report Forms screen where we saw these forms included in the list.

We first produced the LIST report. We returned back to the Primary Option Menu, where we selected 2 to go the Report Set screen. On the Report Set screen, we entered NEW JVML to create a new Report Set. On the EDIT Report Set screen, we entered S next to the List option in the category Performance Reports. On the Performance List Report screen (Figure 12-8), we entered the APPLID and image name as generated by the take-up, specified to use the JVMLST Report Form, and entered a report title.
We returned to the Edit Report Set screen where we ran the List report. We entered the time interval for our report as shown in Figure 12-9 and then submitted the report.

Figure 12-9  Time interval selection for JVML report

Figure 12-10 and Figure 12-11 show the Performance List report.

Figure 12-10  First run: List Report (Part 1 of 2)
In a similar way, we created a Report Set JVMS that uses Report Form JVMSUM. On the Report Set screen, we chose the summary instead of the list. The Summary report produced is shown in Figure 12-12.

These reports show high values for dispatch time, JVM elapsed time and JVM initialization time and extremely high values for response time. The average times in the summary report are consistent with the individual values in the list report so that not a few deviating individual values impact the average values.

Most of the average JVM initialization time is going into the average JVM elapse time. Most of the average JVM elapse time is going into the average dispatch time so that we can say that the JVM initialization time is determining the dispatch time.

To find an explanation for the extreme high response time, we created a new summary Report Form that we called JVMDSP. We moved most dispatch fields that can give information about dispatch delays before the EOR indicator, as shown in Figure 12-13.
When we ran the JVMS Report Set with this form, we received the report shown in Figure 12-14.

From this report, we see that the first dispatch wait time is almost zero which means that there are no problems for the CICS dispatcher to accept and initiate new entered transactions.

Most of the dispatch delay time is spent on MAXJTDly. This means that we are waiting for a JVM TCB to initialize a new JVM. For a better understanding of what is happening, we decided to make one more list form to print the individual transaction information starting from the beginning of our test run. The form was called JVMBEGIN. It contained the fields that distinguish the individual transactions such as terminal ID and task number plus the dispatch and JVM fields that we were using already. We ran the JVML Report Set with the JVMBEGIN Report Form. In the JVML Report Set, we changed the starting time to the time we started our test run. Figure 12-15 shows the JVMBEGIN Report Form.
Figure 12-15 JvMBegin Report Form

Figure 12-16 and Figure 12-17 show the resulting list report.

VIR390
CICS Performance Analyzer
Performance List

LIST0011 Printed at 14:16:12 10/16/2003 Data from 15:41:49 10/14/2003 APPLID SCSJPA6 Page 1
LIST report of JVM related fields

<table>
<thead>
<tr>
<th>Tran Term</th>
<th>TaskNo</th>
<th>Start Time</th>
<th>Stop Time</th>
<th>Response Time</th>
<th>Dispatch Time</th>
<th>MaxJTDly Time</th>
<th>JVMITime</th>
<th>JVM Elap Time</th>
<th>JVM Susp Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTS1 &lt;AY1</td>
<td>105</td>
<td>15:42:00.746</td>
<td>15:42:18.238</td>
<td>17.4924</td>
<td>17.4778</td>
<td>.0000</td>
<td>.0018</td>
<td>.0146</td>
<td>15.5917</td>
</tr>
<tr>
<td>JTS1 &lt;AY7</td>
<td>129</td>
<td>15:42:12.759</td>
<td>15:42:57.271</td>
<td>44.5112</td>
<td>17.1444</td>
<td>.0000</td>
<td>.0044</td>
<td>27.3688</td>
<td>27.3490</td>
</tr>
<tr>
<td>JTS1 &lt;AY1</td>
<td>142</td>
<td>15:42:18.647</td>
<td>15:43:27.161</td>
<td>68.5139</td>
<td>17.3103</td>
<td>.0000</td>
<td>.0040</td>
<td>51.2036</td>
<td>51.1911</td>
</tr>
</tbody>
</table>

Figure 12-16 Individual transaction report from the beginning of the first run (Part 1 of 2)
In this report, in the Suspend Time column, we see that the first five transactions start almost immediately. They have a MAXJTDL Y of zero, MAXJTDL Y being the elapsed time during which the user task waited to obtain a CICS JVM TCB, because the CICS system had reached the limit set by the system parameter, MAXJVMTCBS. The sixth transaction, with task number 113, entered the system at 15:42:04. It could not run immediately because there was no JVM TCB available. It was suspended until the first transaction frees a JVM TCB. The first transaction, with task number 103, stopped at 15:42:15 and has a dispatch time of 16 seconds. Transaction 113 started 5 seconds after transaction 103, so it needed to wait 11 seconds before a JVM TCB became available. This is reflected in the total suspend and the MAXJTDL Y times. We see the next group of five transactions. Transactions 113 to 121 have suspend times between 11 and 14 seconds and the third group, and transactions 123 to 131 have times between 23 and 29 seconds.

TPNS is starting up to 50 transactions. When a transaction returns an output, immediately another is initiated from the same terminal. The first transaction stopped at 15:42:15.790. We can see that TPNS immediately initiated a new transaction, with number 137, from the same terminal, <ATJ, at 15:42:16.196. We initiated some new work before all 50 terminals each can see that TPNS immediately initiated a new transaction, with number 137, from the same terminal. The first transaction stopped at 15:42:15 and has a dispatch time of 16 seconds. Transaction 113 started 5 seconds after transaction 103, so it needed to wait 11 seconds before a JVM TCB became available. This is reflected in the total suspend and the MAXJTDL Y times. We see the next group of five transactions. Transactions 113 to 121 have suspend times between 11 and 14 seconds and the third group, and transactions 123 to 131 have times between 23 and 29 seconds.

In this report, in the Suspend Time column, we see that the first five transactions start almost immediately. They have a MAXJTDL Y of zero, MAXJTDL Y being the elapsed time during which the user task waited to obtain a CICS JVM TCB, because the CICS system had reached the limit set by the system parameter, MAXJVMTCBS. The sixth transaction, with task number 113, entered the system at 15:42:04. It could not run immediately because there was no JVM TCB available. It was suspended until the first transaction frees a JVM TCB. The first transaction, with task number 103, stopped at 15:42:15 and has a dispatch time of 16 seconds. Transaction 113 started 5 seconds after transaction 103, so it needed to wait 11 seconds before a JVM TCB became available. This is reflected in the total suspend and the MAXJTDL Y times. We see the next group of five transactions. Transactions 113 to 121 have suspend times between 11 and 14 seconds and the third group, and transactions 123 to 131 have times between 23 and 29 seconds.

TPNS is starting up to 50 transactions. When a transaction returns an output, immediately another is initiated from the same terminal. The first transaction stopped at 15:42:15.790. We can see that TPNS immediately initiated a new transaction, with number 137, from the same terminal, <ATJ, at 15:42:16.196. We initiated some new work before all 50 terminals each initiated a transaction. This explains why the MAXJTDL Y suspend time is going up slowly. It also explains why it takes time to reach a point where the values of average response time, average dispatch time, and average suspend stabilize to the values as shown in Figure 12-12 on page 276.

### 12.3.2 Xresettable=YES

Knowing that JVM initialization is a time consuming operation, we decided not to increase the MAXJVMTCBS from its default value of 5 to a higher value. Instead we preferred to make a test run with Xresettable=YES in the CICS system properties file in the DFHJVM data set.
Figure 12-18 Second run: List report

Using the JVMS Report Set in combination with the JVMSUM Report Form produced the output shown in Figure 12-19.

For the first two reports, we used the same Report Forms and Report Sets as for the first run. The list produced with the JVML Report Set using the JVMLST Report Form is shown in Figure 12-18.
When we compare the report in Figure 12-19 with the report in Figure 12-12 on page 276, we immediately see the positive effect of resettable JVM. The transaction rate went from 19 transactions per minute to 2675 transactions per minute. Response time dropped from 187 seconds to .38 seconds. We see also that for a reused JVM, the initialization time became zero. Note that we use here a subset of five minutes that excludes the JVM initialization of the first five transactions.

However, in the summary report, we still see that the Avg Suspend Time still makes up the biggest part of the response time. We run again the JVMS Report Set with the JVMDSP Report Form. The result is shown in Figure 12-20.

We see that there is still a delay in availability of JVM TCBs. Our next attempt to improve performance is to specify the ibm.jvm.shareable.application.class.path property in the HFS properties file.

### 12.3.3 Using the sharable application classpath

For this run, we used the same Report Forms and Report Sets as for the previous runs. We only changed the SMF data set name in the systems definition and the date and time interval when submitting the print jobs. Again, we used a time interval of five minutes, not including the start or end of the test period.

Here we show only the output of the summary reports. Figure 12-21 shows the summary report using the sample provided JVMSUM Report Form. This first summary report shows that, by caching the used classes, we could increase the throughput to 3431 transactions per minute. We also see that all measured CPU values decreased as well as almost all elapsed time values.
Figure 12-21  Third run: JVMS + JVMSUM

Figure 12-22 shows the report using the JVMDS Report Form that we created ourselves including the time waiting for a JVM to become available. The second Summary report shows that there is still a 50 milliseconds queuing delay time on the MAXJVMTCBS number of TCBs. In the last run, we decided to see the effect of allocating one more TCB. With the CEMT SET DISPATCH command, we changed the number of MAXJVMTCBS to six and launched a new TPNS workload.

Figure 12-22  Third run: JVMS + JVMDSP

Figure 12-23 and Figure 12-24 show the two summary reports.

Figure 12-23  Final run: JVMS + JVMSUM

The report shows that all values, compared to the previous report, are more or less the same. Only the MAXJTDLY time went down from .0558 to .0487, due to one additional TCB for running JVM programs. Increasing the number of JVM TCBs does not change to the performance of the running JVM program. Given the type of workload introduced by TPNS, where a new transaction is immediately initiated when the previous one terminates, it is difficult to eliminate the delay time on the JVM TCBs as long as the number of TCBs does not reach, or is equal to, the number of simulated terminals in TPNS.
12.4 Conclusion

The CICS PA reports that we produced with this scenario confirm that the initiation of a JVM is an expensive operation. You should avoid it as much as possible. We recommend that you always use a properties file that contains Xresettable=YES.

We also recommend that you use sharable application classpaths to cache your classes and prevent their reload. Our JVM program contained only two classes and we saw a remarkable gain in CPU and elapsed time.
Chapter 13. Enterprise JavaBeans in CICS

This chapter explains how to use CICS Performance Analyzer (PA) to report on Java Virtual Machine (JVM) statistics. We used an application that has three versions, all of them providing exactly the same user interface and functionality. The difference is that the enterprise bean that implements the business logic can use three different interfaces to manage user data.

The three different versions of the enterprise bean are:

- **JCICS version**: Uses JCICS classes to directly access VSAM data sets. We identify this version as JCIM business application and associate JCIM transaction code with the bean name.
- **SQLJ version**: Uses the SQLJ Java application programming interface (API) (static Structured Query Language (SQL)) to access DB2 data. We identify this version as SQLM business transaction and associate SQLM transaction code with the bean name.
- **JDBC version**: Uses the Java Database Connectivity (JDBC) API (dynamic SQL) to access DB2 data. We identify this version as JDBM business transaction and associate JDBM transaction code with the bean name.

We ran three different versions of an enterprise bean application 100 times, each over a single thread to produce CICS PA JVM reports. Then we ran each enterprise bean application 10 times over 10 threads and then five times over 20 threads. We used CICS PA to produce performance reports for each variety of the enterprise bean workload.

A single run of an enterprise bean application contains 13 method calls. In other words, execution of a single business transaction causes execution of 13 CICS transactions.

**Note**: These scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer reports when running with an enterprise bean workload. The CICS regions were not necessarily tuned for peak performance. In some cases, they had a high level of tracing active. Therefore these scenarios and the results provided are for demonstration purposes only. They do not provide definitive results for a customer environment.
13.1 CICS logical EJB server configuration

You can implement a CICS Enterprise JavaBean (EJB) server in a single CICS region. However, in a production environment, you may want to create a server consisting of multiple regions. In such a configuration, a failure of a single region is less critical, and you can implement workload balancing while providing a single system image.

A CICS logical EJB server consists of the following elements:

- A defined set of cloned listener regions that have identical TCPIPSERVICE definitions to listen for incoming Internet Inter-ORB Protocol (IIOP) requests
- A set of cloned application owning regions (AORs), each of which supports an identical set of enterprise beans in identically defined CORBA servers.

Workload balancing can be implemented at two levels:

- Balancing client connections across the listener regions: We used TCP/IP port sharing to achieve this goal.
- Balancing method requests across the AORs: We used the CICSPlex System Manager (SM) provided distributed routing program, EYU9XLOP, to achieve this goal.

For our tests, we implemented the configuration described in 5.1.2, “Configuration for Enterprise JavaBean workloads” on page 130, and shown in Figure 13-1.

For a complete description about how to set up a CICS logical EJB server, refer to CICS Transaction Server for z/OS Java Applications in CICS, SC34-6000, and to Chapter 5 of Enterprise JavaBeans for z/OS and OS/390 CICS Transaction Server V2.2, SG24-6284. You can also refer to Chapters 10, 11, and 12 in SG24-6284 for a description of the ITSO enterprise bean Trader application.
13.2 Single thread enterprise bean scenario

We use CICS PA reports to compare the resources used by the three different enterprise beans. See 5.1.2, “Configuration for Enterprise JavaBean workloads” on page 130, for a description of the system setup that enabled us to run the EJB scenario.

13.2.1 CICS TS V2.2 JDBC/SQLJ support

CICS Java applications can access DB2 data via the JDBC and SQLJ APIs. The JDBC API uses the dynamic SQL model. The SQLJ API uses the static SQL model.

In a CICS environment, the DB2 JDBC driver is link-edited with the CICS DB2 language interface stub DSNCLI. Therefore, JDBC and SQLJ requests are converted by the JDBC driver into EXEC SQL requests and then routed into the existing CICS-DB2 Attachment Facility.

13.2.2 Initiating transaction workload

We associated the following transaction codes with the bean name:

- SQLM for the SQLJ version
- JDBM for the JDBC version
- JCIM for the JCCS (VSAM) version

We used the CREA transaction in CICS to provide REQUESTMODEL definitions for various methods. For each method, we defined a different transaction identifier. This way, we could monitor performance down to the method level. REQUESTMODEL definitions had to be installed in all regions, listener regions as well as AORs.

Figure 13-2 shows the transaction identifiers for the methods for the SQLJ version. We also associated transaction code SOLN with the create() method.

<table>
<thead>
<tr>
<th>SQLM</th>
<th>DistributedTraderAgent</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL1</td>
<td>buy</td>
</tr>
<tr>
<td></td>
<td>long</td>
</tr>
<tr>
<td>SQL2</td>
<td>getCompanyDetails</td>
</tr>
<tr>
<td>SQL3</td>
<td>getCustomerName</td>
</tr>
<tr>
<td>SQL4</td>
<td>listCompanies</td>
</tr>
<tr>
<td>SQL5</td>
<td>listPosition</td>
</tr>
<tr>
<td>SQL6</td>
<td>selectCompany</td>
</tr>
<tr>
<td></td>
<td>java.lang.String</td>
</tr>
<tr>
<td>SQL7</td>
<td>sell</td>
</tr>
<tr>
<td></td>
<td>long</td>
</tr>
<tr>
<td>SQL8</td>
<td>getEJBHome (inherited from EJBObject)</td>
</tr>
<tr>
<td>SQL9</td>
<td>getHandle (inherited from EJBObject)</td>
</tr>
<tr>
<td>SQLA</td>
<td>getPrimaryKey (inherited from EJBObject)</td>
</tr>
<tr>
<td>SQLB</td>
<td>isIdentical (inherited from EJBObject)</td>
</tr>
<tr>
<td></td>
<td>javax.ejb.EJBObject</td>
</tr>
<tr>
<td>SRLR</td>
<td>remove (inherited from EJBObject)</td>
</tr>
</tbody>
</table>

Figure 13-2 SQLJ transactions and methods

Figure 13-3 shows the transactions and method association for the JDBC version. Again we used the CREA transaction to create the REQUESTMODELs and installed them in all regions. We also associated transaction code JDBN with the create() method.
Figure 13-3  JDBC transactions and methods

Figure 13-4 shows the transactions and method association for the JCICS (VSAM) version. Again we used the CREA transaction to create the REQUESTMODELs and installed them in all regions. We also associated the JCIN transaction code with the `create()` method.

Figure 13-4  JVSM transactions and methods

Remember that in logical CICS EJB server, REQUESTMODEL definitions must be installed in IIOP listener regions as well as in AORs.

To produce a workload, we used a batch file that drives the JDBC, SQLJ, and JVSM applications. In this test, we ran each business application sequentially (that is, using only one thread), a total of 100 times.

**Producing CICS Performance Analyzer reports**

We let the batch run of the SQLJ application complete. We flushed the CICS SMF buffers. We did this by turning performance monitoring off and on in both SCSCPJ6 and SCSCPJ7. We then switched the SMF data sets, using the `/I SMF` command. Our SMF data was copied into SMFDATA.ALLRECS.G8684V00. We needed to add this SMF data to the CICS system.
definitions that were used to run the workload, in this case SCSCPJA6 and SCSCPJA7 (Figure 13-5).

Figure 13-5 Adding SMF data to CICS System definition

After we added the SMF data to the CICS System Definitions, we created a group that contained the two CICS System Definitions. To do this, we used option 3 of the System Definitions menu. In Figure 13-6, we specified new EJBGRP to create a new group.

Figure 13-6 New group EJBGRP

On the next screen, we added the two CICS system definitions (SCSCPJA6 and SCSCPJA7) that we wanted to include in the new group (Figure 13-7).

Figure 13-7 Adding CICS system definitions
To obtain the reports that we required for the JVM, we used the sample reports JVMLST (List Report Form) and JVMSUM (Summary Report Form). The sample Report Forms are found on the Report Forms screen, under Samples. After selecting the samples, we populated the Report Forms data set with the sample Report Forms. We did this and had the list of all sample Report Forms. Figure 13-8 shows some of the samples available. We selected JVMLST and JVMSUM.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSUM</td>
<td>SUMMARY</td>
<td>Interval Control Activity</td>
</tr>
<tr>
<td>IMSDBLST</td>
<td>LIST</td>
<td>Transaction DBCTL Usage Analysis</td>
</tr>
<tr>
<td>IMSDBSUM</td>
<td>SUMMARY</td>
<td>Transaction DBCTL Usage Analysis</td>
</tr>
<tr>
<td>IMSRQLST</td>
<td>LIST</td>
<td>Transaction DBCTL Req Analysis</td>
</tr>
<tr>
<td>IMSRQSUM</td>
<td>SUMMARY</td>
<td>Transaction DBCTL Req Analysis</td>
</tr>
<tr>
<td>IMSSUM</td>
<td>SUMMARY</td>
<td>IMS DBCTL PSB Usage Analysis</td>
</tr>
<tr>
<td>JCLST</td>
<td>LIST</td>
<td>Journaling/Logging Activity</td>
</tr>
<tr>
<td>JCSUM</td>
<td>SUMMARY</td>
<td>Journaling/Logging Activity</td>
</tr>
<tr>
<td>JVMLST</td>
<td>LIST</td>
<td>Java Virtual Machine Analysis</td>
</tr>
<tr>
<td>JVMSUM</td>
<td>SUMMARY</td>
<td>Java Virtual Machine Analysis</td>
</tr>
<tr>
<td>PCLST</td>
<td>LIST</td>
<td>Program Request Activity</td>
</tr>
<tr>
<td>PCSUM</td>
<td>SUMMARY</td>
<td>Program Request Activity</td>
</tr>
<tr>
<td>PSTORLST</td>
<td>LIST</td>
<td>Program Storage Analysis</td>
</tr>
<tr>
<td>PSTORSUM</td>
<td>SUMMARY</td>
<td>Program Storage Analysis</td>
</tr>
<tr>
<td>RMIDBLST</td>
<td>LIST</td>
<td>CICS RMI Analysis - DB2 Overview</td>
</tr>
<tr>
<td>RMIDBSUM</td>
<td>SUMMARY</td>
<td>CICS RMI Analysis - DB2 Overview</td>
</tr>
</tbody>
</table>

Figure 13-8  Sample Report Forms

After we set the forms we wanted, we created a Report Set to use for the performance report. To create a Report Set, we selected option 2 on the CICS Performance Analyzer main menu. We specified new EJBREP and created the new Report Set (Figure 13-9).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2REPS</td>
<td>CICS PA Report Set</td>
<td>2003/10/14 12:00</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>DB2REP2</td>
<td>CICS PA Report Set</td>
<td>2003/10/15 13:35</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>EXCREP</td>
<td>CICS PA Report Set</td>
<td>2003/10/23 11:28</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>LOGREPS</td>
<td>CICS PA Report Set</td>
<td>2003/10/22 14:39</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>LOGREP2</td>
<td>CICS PA Report Set</td>
<td>2003/10/18 09:48</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>REPORT7</td>
<td>CICS PA Report Set</td>
<td>2003/10/25 12:09</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>REPORT1</td>
<td>CICS PA Report Set</td>
<td>2003/10/23 08:49</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>REPORT2</td>
<td>CICS PA Report Set</td>
<td>2003/10/25 11:05</td>
<td>CICLSLS5</td>
</tr>
<tr>
<td>REPORT3</td>
<td>CICS PA Report Set</td>
<td>2003/10/22 18:47</td>
<td>CICLSLS5</td>
</tr>
</tbody>
</table>

Figure 13-9  New Report Set
The two sample Report Forms were added to the list of Report Form that we already defined (see Figure 13-10). After the Report Forms were added, they did not appear in the list of samples.

![Figure 13-10 Report Forms](image)

We used the Report Set to produce both a List and a Summary report for transaction SQLM. We selected List in the category Performance Reports (Figure 13-11).

![Figure 13-11 Specifying the List Performance Report](image)

On the Performance List Report screen (Figure 13-12), we specified JVMLST in the Form field. We also chose Performance for selection criteria.
On the Performance Select Statement screen, we selected transaction SQL* (Figure 13-13). This is because we wanted all the transaction codes from all the methods that made up the SQLM transaction.

After we finished with the List reports, we saved our changes and selected the Summary Report, used the JVMSUM form, and selected the transaction SQL*. We ran both the List and Summary reports. We needed to add the group EJBGRP to the Run Report Set screen (Figure 13-14).
Figure 13-14  Run Report Set

Figure 13-15 shows part of the first page from the list report. It shows one SQLM transaction and some of its method transactions.

<table>
<thead>
<tr>
<th>Tran Userid</th>
<th>TaskNo</th>
<th>Stop Time</th>
<th>Response Time</th>
<th>Dispatch Time</th>
<th>User CPU</th>
<th>KY8 CPU</th>
<th>JB CPU</th>
<th>JVM Elap Time</th>
<th>JVM Time</th>
<th>JVM Susp Time</th>
<th>JVM Meth</th>
<th>JVMR Time</th>
<th>JVM Susp</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLM CICUSER</td>
<td>164</td>
<td>17:16:44.230</td>
<td>5.6586</td>
<td>5.5701</td>
<td>1.0281</td>
<td>1.0270</td>
<td>1.0248</td>
<td>5.6576</td>
<td>.0000</td>
<td>5.5813</td>
<td>.0763</td>
<td>.0884</td>
<td></td>
</tr>
<tr>
<td>SQL4 CICUSER</td>
<td>165</td>
<td>17:16:46.322</td>
<td>2.1161</td>
<td>2.1078</td>
<td>.4671</td>
<td>.4661</td>
<td>.4624</td>
<td>2.1145</td>
<td>.0000</td>
<td>2.0477</td>
<td>.0668</td>
<td>.0082</td>
<td></td>
</tr>
<tr>
<td>SQL6 CICUSER</td>
<td>166</td>
<td>17:16:46.905</td>
<td>.5870</td>
<td>.5801</td>
<td>.3266</td>
<td>.3256</td>
<td>.3215</td>
<td>.5849</td>
<td>.0000</td>
<td>.5475</td>
<td>.0374</td>
<td>.0065</td>
<td></td>
</tr>
<tr>
<td>SQL3 CICUSER</td>
<td>167</td>
<td>17:16:48.967</td>
<td>.1432</td>
<td>.1367</td>
<td>.1176</td>
<td>.1167</td>
<td>.1157</td>
<td>.1421</td>
<td>.0000</td>
<td>.1369</td>
<td>.0052</td>
<td>.0063</td>
<td></td>
</tr>
<tr>
<td>SQL5 CICUSER</td>
<td>168</td>
<td>17:16:49.313</td>
<td>.3188</td>
<td>.3123</td>
<td>.1503</td>
<td>.1494</td>
<td>.1459</td>
<td>.3171</td>
<td>.0000</td>
<td>.3106</td>
<td>.0064</td>
<td>.0064</td>
<td></td>
</tr>
<tr>
<td>SQL5 CICUSER</td>
<td>169</td>
<td>17:16:49.482</td>
<td>.1332</td>
<td>.1274</td>
<td>.0885</td>
<td>.0876</td>
<td>.0842</td>
<td>.1315</td>
<td>.0000</td>
<td>.1263</td>
<td>.0052</td>
<td>.0057</td>
<td></td>
</tr>
<tr>
<td>SQL1 CICUSER</td>
<td>170</td>
<td>17:16:49.683</td>
<td>.1631</td>
<td>.1562</td>
<td>.0695</td>
<td>.0685</td>
<td>.0634</td>
<td>.1614</td>
<td>.0000</td>
<td>.1558</td>
<td>.0056</td>
<td>.0068</td>
<td></td>
</tr>
<tr>
<td>SQL7 CICUSER</td>
<td>171</td>
<td>17:16:49.944</td>
<td>.2274</td>
<td>.2204</td>
<td>.1286</td>
<td>.1257</td>
<td>.1207</td>
<td>.2257</td>
<td>.0000</td>
<td>.2205</td>
<td>.0052</td>
<td>.0068</td>
<td></td>
</tr>
<tr>
<td>SQLM CICUSER</td>
<td>172</td>
<td>17:16:50.154</td>
<td>.1789</td>
<td>.1726</td>
<td>.0765</td>
<td>.0755</td>
<td>.0745</td>
<td>.1781</td>
<td>.0000</td>
<td>.1490</td>
<td>.0291</td>
<td>.0062</td>
<td></td>
</tr>
<tr>
<td>SQL3 CICUSER</td>
<td>173</td>
<td>17:16:50.326</td>
<td>.1572</td>
<td>.1508</td>
<td>.1303</td>
<td>.1294</td>
<td>.1285</td>
<td>.1561</td>
<td>.0000</td>
<td>.1510</td>
<td>.0051</td>
<td>.0063</td>
<td></td>
</tr>
<tr>
<td>SQL5 CICUSER</td>
<td>174</td>
<td>17:16:50.577</td>
<td>.2200</td>
<td>.2137</td>
<td>.1776</td>
<td>.1767</td>
<td>.1734</td>
<td>.2184</td>
<td>.0000</td>
<td>.2131</td>
<td>.0053</td>
<td>.0061</td>
<td></td>
</tr>
<tr>
<td>SQL5 CICUSER</td>
<td>175</td>
<td>17:16:50.708</td>
<td>.0981</td>
<td>.0918</td>
<td>.0717</td>
<td>.0708</td>
<td>.0675</td>
<td>.0965</td>
<td>.0000</td>
<td>.0904</td>
<td>.0061</td>
<td>.0062</td>
<td></td>
</tr>
</tbody>
</table>

Figure 13-15  Performance List report
The Performance Summary Report follows the Performance List Report (Figure 13-16).

Figure 13-16  Performance Summary report

Figure 13-17 shows the elapsed time of the SQLM workload. From this report, you can see that running all 100 transactions took 5 minutes and 50 seconds. That is, we achieved a transaction rate of 4 transactions per second (1300 CICS transactions in 350 seconds).

Figure 13-17  Elapsed time for SQLM transactions
We ran the JDBM transaction and collected the SMF data in the same way as for the SQLM transaction. Now, we did not need to set up a new Report Form since we already did so for the SQLM transaction. However, we needed to add the SMF data set to the CICS APPLIDs on the System Definition screen. We wanted to create a new Report Set for the JDBM transaction. We did this by specifying new EJBREP2 on the Report Set screen (Figure 13-18).

![Figure 13-18  New Report Set]

On the new Report Set, we selected both the List and Summary report options. As shown in Figure 13-19, we added the Report Form JVMLST to the Performance List report. We also selected the Performance selection criteria.

![Figure 13-19  List Report]
In the Selection Criteria screen, we requested to include just the JDB* transactions (Figure 13-20).

```
Command ===>

Active  --------------- Report Interval ---------------
Inc Start  --------- From ------------------------ To ----------------

Inc Field           --- Value or Range --- Object
/ Exc Name +         Type    Value/From To       List +
INC TRAN            JDB*
```

**Figure 13-20  Selection criteria**

We set up the Summary Report with the same selection criteria option and added the form JVMSUM. We ran the new reports. Both the List and Summary Performance Reports were active as indicated by yes in the Active column for each. We entered a RUN command on the command line to run both reports at the same time. The group EJBGRP was still specified on the Report Set screen, so we did not need to add it again (Figure 13-21).

```
Command ===>

File Systems Options Help
Run Report Set EJBREP2

Specify run Report Set options then press Enter to continue submit.

System Selection:
CICS APPLID . . + Image . . + Group . . EJBGRP +

/ Override System Selections specified in Report Set

Missing SMF Files Option:
1 1. Issue error message From
2. Leave DSN unresolved in JCL To
3. Disregard offending reports

Enter "/" to select option
/ Edit JCL before submit
```

**Figure 13-21  Submitting Report Set**
Because we used workload balancing of method requests, we needed to tie together transactions from both SCSCPJA6 and SCSCPJA7 for a complete picture. The first transaction is JDBN (create) and it is task number 1428. The next task number was 4564. These transactions ran on the different AORs.

We looked at the Performance Summary report (Figure 13-23). The JDBN transaction that corresponds to the bean create() method was run 100 times.
Figure 13-24 shows the elapsed time of the JDBM workload. The throughput of the 100 JDBM business transactions was completed in five minutes or a transaction rate of 4 CICS transactions per second (1300 CICS transactions in 300 seconds).

![Figure 13-24  Elapsed time for JDBM workload](image)

We then ran the JVSM transaction workload using a batch workload. We collected the SMF records the same as before, and added the records to the System Definitions for SCSCPJA6 and SCSCPJA7. The first action was to create a new Report Set, called EJBREP3. Again we used the same sample Report Forms JVMLST and JVMSUM, as we used for the SQLJ and JDBC reports. We needed different selection criteria, all transactions JCI* (Figure 13-25).

![Figure 13-25  JVSM selection criteria](image)
Figure 13-26 shows part of the report.

We submitted the new Report Set. Figure 13-26 shows part of the report.

Figure 13-26  JVSM List Report

We submitted the new Report Set. Figure 13-26 shows part of the report.

Figure 13-27 is the Summary Report for the JVSM transaction run.

Figure 13-27 - JVSM Summary Report
Figure 13-28 shows the elapse time for the JCICS VSAM workload.

The JVSM workload completed in 4 minutes and 40 seconds or a transaction rate of 5 per second (1300 CICS transactions in 280 seconds).

This scenario demonstrated the use of CICS Performance Analyzer in providing JVM reports for the different enterprise beans. The results were not surprising, since we used only one thread. In the next scenario, we produce a workload on a greater number of threads to see if we can achieve a better performance.

13.3 Multithread enterprise bean scenario

This section describe an enterprise bean performance scenario that is running several EJB clients concurrently. We used the SQLJ, JDBC, and JCICS (VSAM) version of the DistributedTraderAgent enterprise bean to produce relevant performance reports of each enterprise bean application workload.

We used CICSPlex SM Workload Management to balance method call invocations across our AOR regions PJA6 and PJA7. CICSPlex SM is using its distributed routing program EYU9XLOP to dynamically route each single method of the DistributedTraderAgent enterprise bean to the least loaded AOR. A client method call invocation to the stateful DistributedTraderAgent enterprise bean never runs under an existing OTS transaction since we defined a transaction attribute of NotSupported. Therefore, each method call invocation is routed by CICSPlex SM to the most efficient AOR. Refer to CICS Transaction Server for z/OS Java Applications in CICS, SC34-6000, for more information about enterprise bean applications in CICS.

See 13.2.2, “Initiating transaction workload” on page 287, where we explain how we associated transaction IDs with enterprise bean methods of the SQLJ, JDBC, and JCICS (VSAM) EJB applications. To measure the performance of enterprise beans in CICS, we monitored the performance data of the IIOP request receiver and request processor transactions. We used three sets of cloned CIRP transactions to monitor each single method call invocation through an associated transaction ID.

13.3.1 Workload generation

We created a Java front-end application (Figure 13-29) that allows to run the Java enterprise bean workload using parallel threads on a client workstation rather than a single threaded sequence of client method calls. The program starts the Windows command files that we used to generate the workload for the sequential enterprise bean scenario.
Each command file runs in its own thread, which simulates clients that run the enterprise bean application concurrently. The Java front-end program uses three parameters:

- The name of the command file: For this performance scenario, we used the following command files:
  - `SQLJ.cmd` runs the client of the DistributedTraderAgent enterprise bean that is using the SQL API to access its database
  - `JDBC.cmd` runs the client of the DistributedTraderAgent enterprise bean that is using the JDBC API to access its database
  - `JVSM.cmd` runs the client DistributedTraderAgent enterprise bean that is using the JCICS VSAM classes to access its data sets

- The number of threads that should be created
- The number of calls per thread

AORs PJA6 and PJA7 are configured to use 10 JVMs each. Therefore, we decided to run two different workloads per enterprise bean application. For the first run, we executed our Java program and specified the following values:

- SQLJ.cmd
- 10
- 10

We created ten threads. Each thread called the SQLJ.cmd file ten times, which resulted in a total of 100 calls to the SQLJ version of the DistributedTraderAgent enterprise bean. This, in fact, is the same number of calls that we used during the sequential enterprise bean performance scenario. To use more of the available JVMs, we started another workload that used 20 threads and five calls per thread. Again this resulted in a total of 100 calls to the DistributedTraderAgent enterprise bean. We specified the following values:

- SQLJ.cmd
- 20
- 5
We ran both workloads using these above parameters for the SQLJ, JDBC, and JCICS (VSAM) version of the enterprise bean application.

### 13.3.2 Objectives

We wanted to demonstrate the performance behavior of CICS enterprise bean applications using SQLJ/JDBC in comparison with the same enterprise bean application using JCICS(VSAM). We defined a CICSPlex SM workload specification for distributed routing specifying the default (QUEUE) algorithm. We expected CICSPlex SM to do dynamic distributed routing of method call invocations to the AOR region that has the shortest execution queues.

We calculated the time that was needed to complete the workload of 100 calls to the DistributedTraderAgent enterprise bean. We used CICS Performance Analyzer reports to show the enterprise bean performance when using 10 or 20 threads to complete the workload.

### 13.3.3 Running the SQLJ workload

We started the workload that calls the SQLJ version of the DistributedTraderAgent enterprise bean using 10 threads and 10 calls per thread. The workload finished after about 1.5 minutes. To collect all relevant SMF records, we flushed the CICS monitoring buffers and issued an /I SMF command to switch SMF data sets. When the SMF data set was archived, we performed the following steps to create the performance summary reports shown in Figure 13-30 and Figure 13-31:

1. Use the Take-Up for SMF function to update AORs PJA6 and PJA7.
2. Create a new Report Set in order to create a summary report.
3. Use sample Report Form JVMSUM to format relevant performance fields for JVMs.
4. Specify a Performance select statement to include a report interval that covers the test interval.
5. Run the report.

When we looked at the reports shown in Figure 13-30 and Figure 13-31, we found that CICSPlex SM routed more transactions to AOR PJA7.

The EJB client calls the DistributedTraderAgent enterprise bean 13 times. To monitor the request stream and request processor transactions, we associated a transaction ID to each method of the bean. One client performs 13 method calls multiplied by 100 clients, resulting in 1300 transactions per workload. AOR PJA6 executed 383 transactions when PJA7 processed 1017 transactions.
When the workload completed, we entered the CEMT I JVM command. We found that PJA6 was using seven JVMs while PJA7 was just using three JVMs. We assumed that PJA6 was busy building additional serially reusable JVMs which lead to longer transaction queues and response times. CICSPlex SM took that into account for its routing decision and started to route more workload to PJA7. CICSPlex SM does not spread the workload evenly. The entire workload could be routed to one CICS system in a CICSpex if it stayed within its goals and continued being the most efficient CICS region. If the workload made 500 calls to the enterprise bean rather than 100 calls, then PJA6 would probably catch up to PJA7 quickly.

We started the same workload again using 20 threads and five calls per thread, which again started 100 calls to the SQLJ version of the DistributedTraderAgent enterprise bean. We repeated the steps that we described earlier to create a performance summary report for PJA6 and PJA7 (Figure 13-32 and Figure 13-33).
When we looked at the performance reports that we created, we found that response times did not improve at all. AOR PJA6 processed 223 transactions while AOR PJA7 processed 1077 transactions. We checked the number of JVMs that we created and found that PJA6 was using ten JVMs, which is the maximum. PJA7 was using five JVMs. Therefore, the average response time of PJA6 was influenced by method call invocations that were suspended waiting for JVM initialization. After the serially reusable JVM was built, response times improved dramatically.

We used a performance list report (Figure 13-34 and Figure 13-35) to select the first and last task number of the workload. Then we used the time stamps to calculate the time that the workload needed to complete. Using 10 threads took 78 seconds to complete 1300 method call invocations. That is an average of 17 transactions per second. The workload that was using 20 threads took 5 seconds longer, which is 83 seconds. This is a throughput of 17 transactions per second.
13.3.4 Running the JDBC workload

To see a comparison with JDBC and SQLJ versions of the DistributedTraderAgent enterprise bean, we started another workload using the Java program that we created earlier. We specified the following values:

- `JDBC_WLM.cmd`
- 10
- 10

This setup again starts the client code for the JDBC enterprise bean 100 times in total. We used 10 threads using 10 calls per thread, which simulated 10 EJB clients running concurrently.

When the workload finished, we flushed the CICS monitoring buffers and switched SMF data sets using the `/I SMF` command. When the SMF data set was archived, we performed the following steps to create a performance summary report for both AORs PJA6 and PJA7:

1. Use the Take-Up from SMF function to update the system definition with new SMF data.
2. Create a new system group for both AORs PJA6 and PJA7.
3. Reuse the Report Set that you used for the SQL workload.
4. Edit the report interval information in the Performance select statement.
5. Run the report.

We used the performance list report to select the first and last task number of the workload (Figure 13-36 and Figure 13-37).
We used the time stamps to calculate the time that the workload needed to complete (Figure 13-38 and Figure 13-39). Using 10 threads took 78 seconds to complete 1300 method call invocations. That is an average of 17 transactions per second. The workload that used 20 threads took 75 seconds. This is the throughput of 17 transactions per second.

**Figure 13-36** Both AORs: 10 threads 10 calls per thread

**Figure 13-37** Both AORs: 20 threads 5 calls per thread
13.3.5 Running the JCICS (VSAM) workload

We started the JCICS (VSAM) version of the DistributedTraderAgent enterprise bean. We started another workload using the Java program that we created earlier. We specified the following values:

- JVSMD_WLM.cmd
- 10
- 10

This started the client code for the JCICS (VSAM) enterprise bean 100 times in total. We used 10 threads, with 10 calls per thread, simulating 10 EJB clients running concurrently.

When the workload finished, we flushed the CICS monitoring buffers and switched SMF data sets using the /I SMF command. When the SMF data set was archived, we performed the following steps to create a performance summary report for both AORs PJA6 and PJA7:

1. Use the Take-Up from SMF function to update the system definition with new SMF data.
2. Create a new system group for both AORs PJA6 and PJA7.
3. Reuse the Report Set we were using for the SQL workload.
4. Edit the report interval information in the Performance select statement.
5. Run the report.

We used the Performance List report to select the first and last task number of the workload (Figure 13-40 and Figure 13-41).

---

**Figure 13-39**  Transactions per second: 10 threads

**Figure 13-40**  Both AORs: 10 threads, 10 calls per thread
We used the time stamps to calculate the time that the workload needed to complete. Figure 13-42 shows that the JVSM transaction workload took 1 minutes and 11 seconds to complete when running with 10 threads or a transaction rate of 18 per second.

We then ran a mixed workload consisting of SQLJ, JDBC and JVSM transactions. We used 10 threads and ran 10 transactions of each enterprise bean on each thread, for total of 300 business transactions or, when broken down into method call transactions, a total of 3900.

To produce the CICS Performance Analyzer report for this run of the mixed transactions, we wanted to create an Object List containing the three types of transactions. We called the Object List EJBTRAN and added the transactions that we wanted to select. Figure 13-44 shows the selections we made.
We added the Object List to the Performance select statements for both the List and Summary reports. Figure 13-45 shows the Object List being used in the Performance Select Statement screen.

Figure 13-44 Object List selection

Figure 13-45 Performance Select Statement screen
## 13.4 Conclusion

CICS Performance Analyzer proved useful in providing reports on the use of JVMs in a CICS environment. From the CICS PA reports, in these runs, the JVSJ transaction performed slightly better than the SQLJ and JDBC transactions. However, we did not see noticeable difference in performance between SQLJ and JDBC versions of our application. The reason may be that we did not customize the SQLJ default serializable profiles on z/OS, assuming that it was done in an earlier ITSO project when the application was developed. By default, if you run an SQLJ program with an uncustomized profile, SQL calls are mapped to pure JDBC calls at run time. Since there is no easy way to check whether the JAR file contains customized or uncustomized profiles, we decided not to investigate this issue further.
Application Naming support

Application Naming support allows you to exercise greater control over the identification and collection of performance data. It allows more granularity over a “transaction ID” since it allows you to specify an alternate transaction ID. This way you can relate individual transactions into a single application name or to separate different linked programs that run under one common transaction ID as different transactions.

A good reason to do so can be an application that starts from one common menu. Depending on the selected option, a different program is run. On entry into each program, you can assign a different alternate transaction code for the execution of this program. With CICS Performance Analyzer (PA), you can select this alternate transaction code instead of using the common transaction ID.

This chapter discusses the Application Naming support introduced by CICS Transaction Server (TS) V2.2 authorized program analysis report (APAR) PQ63143. The equivalent CICS TS V1.3 APAR is PQ63141. For a description of Transaction Resource Monitoring support also introduced by these APARs, refer to 6.7, “Transaction Resource Monitoring” on page 163. For the complete list of enabling program temporary fixes (PTFs), refer to Table 2-1 on page 26.
14.1 Implementing Application Naming support

A new parameter, APPLNAME={NO | YES}, is added to the DFHMCT TYPE=INITIAL macro. The APPLNAME option specifies whether you want to use the application support provided by CICS monitoring. This is an enabling function that allows application programs to invoke special CICS event monitoring points called DFHAPPL.1 and DFHAPPL.2.

When you specify APPLNAME=YES, two macros (see Example 14-1) are internally generated.

Example 14-1  Internal generated event monitoring points (EMPs) with APPLNAME=YES

| DFHMCT TYPE=EMP,CLASS=PERFORM,ID=(DFHAPPL.1), | X          |
| PERFORM=(MOVE(0,4)),FIELD=(1,APPLNAME)          | @BA63143   |
| DFHMCT TYPE=EMP,CLASS=PERFORM,ID=(DFHAPPL.2), | X          |
| PERFORM=(MOVE(4,8))                             | @BA63143   |

There is one user event monitoring point field with name APPLNAME added to the system management facility (SMF) record. The SMF record field contains 12 bytes. Two monitoring points, DFHAPPL.1 and DFHAPPL.2, perform the following default operation when invoked:

- **DFHAPPL.1**: Moves four characters from the address specified on the DATA1 option of the EXEC CICS MONITOR command to an offset zero in the SMF field.
- **DFHAPPL.2**: Moves eight characters from the address specified on the DATA1 option of the EXEC CICS MONITOR command to an offset four in the SMF field.

Unlike EMPs which you define explicitly with your own EMP IDs, data moved by invoking the APPLNAME EMPs is not reset by CICS after you force the writing of an SMF record by using the DELIVERY option. However, your application can set different values by invoking the APPLNAME EMPs again.

CICS Performance Analyzer implements these changes by splitting the 12-byte SMF field into two fields, a four-byte field called APPLTRAN and an eight-byte field called APPLPROG. These fields can be requested in Report Forms or the FIELDS/BY command operands. APPLTRAN and APPLPROG are new Selection Criteria fields that allow filtering of reports by Application Naming transaction ID and program.

14.2 Application Naming sample

This section provides a sample EMP, the monitoring control table (MCT), and the results of running the program.

14.2.1 Sample program

To show how you can use the DFHAPPL EMPs, we developed the program shown in Example 14-2. The basic flow of this program is:

1. Write actual transaction code and program name to SMF record in DFHAPPL provided field.
2. Write the date and time to SMF record in its own generated EMP2 field. This EMP also updates a counter.
3. Update the counter again and force the writing of the SMF record.
4. Update the counter twice.
5. Update counter again and force the writing of a second SMF record.
6. Change the values in the DFHAPPL provided fields.
7. Stop the program so that a third SMF record is written.

Example 14-2 Sample program to show DFHAPPL EMP usage

WORKING-STORAGE SECTION.
01   DFHAPPL          PIC X(8)  VALUE 'DFHAPPL'.
01   APPLDATA-PTR     POINTER.
01   APPLTRAN         PIC X(4).
01   APPLPROG         PIC X(8).
01   APPLDATA-LEN     PIC S9(8) COMP VALUE 0.
*  
01   CICS-TIME        PIC X(08) VALUE SPACES.
01   EMP2P            PIC X(8)  VALUE 'EMP2'.
01   PGMTIME-PTR      POINTER.
01   PGMTIME.
 03   CURRENT-DATE PIC X(10).
 03   FILLER       PIC X(01)   VALUE SPACES.
 03   CURRENT-TIME PIC X(08).
01   PGMTIME-LEN      PIC S9(8) COMP VALUE 19.
*  
01   ITSOTRAN         PIC X(4)  VALUE 'ITSO'.
01   ITSOPROG         PIC X(8)  VALUE 'ITSOPROG'.
*  
PROCEDURE DIVISION.
BEGIN.

* OBTAIN TRANSACTION ID FROM EIB AND USE E.C.ASSIGN
* FOR PROGRAM NAME.
*  
MOVE EIBTRNID TO APPLTRAN.
EXEC CICS ASSIGN PROGRAM(APPLPROG)
END-EXEC.
*  
* WRITE TO DFHAPPL EMP.
*  
SET APPLDATA-PTR TO ADDRESS OF APPLTRAN.
EXEC CICS MONITOR POINT(1) ENTRYNAME(DFHAPPL)
DATA1(APPLDATA-PTR) DATA2(APPLDATA-LEN)
END-EXEC.
*  
* SET APPLDATA-PTR TO ADDRESS OF APPLPROG.
EXEC CICS MONITOR POINT(2) ENTRYNAME(DFHAPPL)
DATA1(APPLDATA-PTR) DATA2(APPLDATA-LEN)
END-EXEC.
*  
* OBTAIN DATE AND TIME
*  
EXEC CICS ASKTIME
  ABSTIME(CICS-TIME)
END-EXEC.
EXEC CICS FORMATTIME
  ABSTIME(CICS-TIME)
  DDMMYYYY(CURRENT-DATE)
14.2.2 Monitoring control table

Example 14-3 shows the part of the MCT that we wrote to contain the EMPs as referenced in the program.
Example 14-3   MCT part for Application Naming sample

I1       DFHMCT TYPE=INITIAL,                                          *
          APPLNAME=YES,                                           *
          SUFFIX=I1                                              *

          DFHMCT TYPE=EMP,                                              *
          CLASS=PERFORM,                                         *
          ID=(EMP2.1),                                            *
          COUNT=(1,COUNTER),                                       *
          FIELD=(1,PGMTIME),                                       *
          PERFORM=(ADDCNT(1,1),MOVE(0,19))                          *

          DFHMCT TYPE=EMP,                                              *
          CLASS=PERFORM,                                         *
          ID=(EMP2.2),                                            *
          PERFORM=(ADDCNT(1,1),DELIVER)                           *

          DFHMCT TYPE=EMP,                                              *
          CLASS=PERFORM,                                         *
          ID=(EMP2.3),                                            *
          PERFORM=(ADDCNT(1,1))                                      *

          DFHMCT TYPE=FINAL                                           END

In this MCT, APPLNAME=YES is in the DFHMCT TYPE=INITIAL macro that results in the automatic generation of the APPLNAME field for the DFHAPPL EMP.

The fields that we added ourselves are:

- **EMP2.1**: This EMP defines two fields. The first field is a counter field with name COUNTER that is incremented by one every time we execute an EXEC CICS MONITOR POINT(1) command referencing this EMP. The second field is a character field with the name PGMTIME. Nineteen bytes of data are moved to this field while updating the counter when you run the EXEC CICS MONITOR POINT(1) command.

- **EMP2.2**: This field acts on the same counter that is defined in EMP2.1. When executing an EXEC CICS MONITOR POINT(2) command, the counter is incremented by 1 and the writing of an SMF record is requested through the DELIVERY option.

- **EMP2.3**: This field also acts on the same counter as defined in EMP2.1. Running an EXEC CICS MONITOR command on this EMP only increments the counter by one with no other action.

### 14.2.3 Program run results using the Performance List report

To see the results of a single program run, from CICS PA, we selected the existing SCSCPJA6 definition to update it. We only changed the title and the new SMF data set name. Next, we moved the cursor under Dictionary and entered 1 to create a new dictionary record in the specified dictionary data set. Figure 14-1 shows the Systems Definition screen with those updates.
Then we created a new Report Form. On the Primary Option Menu, we selected 3. The first Report Forms screen is displayed. On this screen, we entered NEW DFHAPPL to create a form with name DFHAPPL. In the pop-up screen, the APPLID was already filled in. We entered 1 to select a list form and pressed Enter. On the EDIT screen, we moved the fields APPLTRAN, APPLPROG, COUNTER, PGMTIME, and START before the EOR indicator. We also requested only those fields together with TRAN, PROG and TASKNO, and CPU. Figure 14-2 shows this Report Form definition.

We saved this Report Form and, in the Primary Option Menu, selected option 2 to create a new Report Set with the same name DFHAPPL. On the Report Sets main screen, we entered NEW DFHAPPL. On the resulting screen, we entered S next to List in the Performance Reports.
category. On the subsequent screen, we entered the APPLID, the Report Form name, and title. Then under the Selection Criteria heading, we selected the Performance option as shown in Figure 14-3.

![Figure 14-3 DFHAPPL Performance List Report](image)

As a result of selecting the Performance option, we received the Performance Select Statement screen (Figure 14-4). On this screen, we chose to include only those SMF records in which the field name TRAN is equal to APPL.

![Figure 14-4 DFHAPPL Performance Select Statement screen](image)

We pressed F3 to save our settings until we returned to the Report Set screen. On the Report Set screen, Global and List have Yes in the active column to indicate that these are the active options. On this screen, we entered the RUN command to run the report that generates the job to print our requested list. When the batch job finished, in the spool, we found the output shown in Figure 14-5.
This output shows that we have three SMF records for task number 5515. During the one time execution of our test program, we first moved the actual transaction code from the EIB and the actual program name, that we acquired through an EXEC CICS ASSIGN PROGRAM command, to the DFHAPPL Tran and Program fields. Then we obtained date and time which we moved to our own defined EMP. At the same time, we set our counter to 1.

The next EXEC CICS MONITOR command was to increment our counter a second time and to force the writing of the SMF record by using the DELIVERY option. This record is printed in the first line of our report. The first Tran and Program fields are the fields as inserted by CICS. Then we see the task number followed by the Tran and Program fields from the DFHAPPL EMP. Our counter is set to 2 and the date and time as we move in the SMF record are present. The Start Time and User CPU Time are again CICS provided fields. They show the start time of the transaction and the amount of CPU time used up to when the SMF record is written.

After this first SMF record was written, all values for all user EMPs are reset. This is different for the DFHAPPL EMP which is not reset and keeps its value. To show this, we executed monitoring point 3 of our own EMP2 twice. This results in the counter becoming 2. We then executed monitoring point 2 again, which brought our counter to a value of 3 and again forced the writing of the SMF record. This is shown in the second line of the report. The counter indeed has a value of 3 and the PGMTIME field is blanked out. Notice also that the Start Time field is the time when the second SMF record was created. The CPU time used since the creation of the SMF record up to the time of writing it, is so low that no value shows up in the field.

To show that the DFHAPPL EMP fields can be modified at any time, we now moved the values ITSO and ITSOPROG to the actual SMF record. This record is written at task end and is the third and last record written for this transaction run. We see that the CICS provided values for Tran and Program are left untouched but the values of the DFHAPPL EMP contain the new values we moved to them.

To summarize, we showed you the new Application Naming feature introduced in CICS TS V2.2 and CICS TS V1.3. We wrote a sample program that shows the basic possibilities of this new feature. To show you how you can externalize this using CICS PA, we produced a performance list report. However, since you can use the newly added fields in every selection criteria, you can use these fields in other reports as well.
CALL and LINK performance

This chapter shows how you can perform performance measurements on an individual transaction level base. It looks at performance issues associated with use of the EXEC CICS LINK command as opposed to the CALL command in CICS COBOL programs. It shows how you can use event monitoring points to add user fields to standard CICS Monitoring Facility (CMF) performance records. Plus, this chapter shows how CICS Performance Analyzer (PA) can handle these user fields.

**Note:** These scenarios were used to provide situations that allow us to demonstrate the use of CICS Performance Analyzer reports. The CICS regions were not necessarily tuned for peak performance. Therefore, these scenarios and the results provided are for demonstration purposes only. They do not provide definitive results for a customer environment.
15.1 Performance testing

This example first shows how you can define your own EMPs and monitoring points to do some performance testing in your own applications. We look at the overhead of the Language Environment® (LE) 370 run-time parameter CBLPSHPOP. We also look at the performance gains obtained by specifying RUNAPPOOL=YES.

Then, we compare the CALL and EXEC CICS LINK commands. For all these exercises, you see how CICS Performance Analyzer can be helpful to produce similar reports on different runs with different data.

15.1.1 What is CBLPSHPOP?

A CICS program can issue an EXEC CICS IGNORE or EXEC CICS HANDLE command. When this program performs an EXEC CICS LINK to another CICS program, the handle effects are not inherited by the LINKed-to program. The LINKed-to program can itself issue the EXEC CICS IGNORE or EXEC CICS HANDLE commands. Upon EXEC CICS RETURN to the first program, the LINKed-to program's HANDLE environment is ignored and the first program continues within its own HANDLE environment. This is possible because internally, CICS keeps the HANDLE environment of the first program after an EXEC CICS LINK.

This behavior becomes different with Language Environment where programs can CALL each other. With a CALL, we do not go through CICS program control so that the HANDLE environment is not saved. If a called program executes its own EXEC CICS HANDLE, it destroys the settings of the first program. Upon return, the calling program inherits the settings of the called program and eventually runs with a HANDLE environment of which it is not aware.

To avoid this, COBOL2 run-time support always executed an EXEC CICS PUSH HANDLE command to save the current effect of EXEC CICS HANDLE and EXEC CICS IGNORE commands. Upon return, an EXEC CICS POP HANDLE command was always executed to restore the caller's environment.

Language Environment run-time support introduced the CBLPSHPOP parameter to make the execution of the EXEC CICS PUSH HANDLE and the EXEC CICS POP HANDLE commands optional. For called programs that do not execute their own EXEC CICS HANDLE or EXEC CICS IGNORE commands or for called programs that issue their own EXEC CICS PUSH HANDLE and EXEC CICS POP HANDLE commands, you can set the CBLPSHPOP parameter to OFF to avoid the overhead of these commands.

15.1.2 CLER transaction

Language Environment provides run-time options to control CICS program's processing. There are four levels at which you can specify run-time options:

- **Installation wide**: The installation-wide options for CICS are compiled in CSECT CEEDOPT in the load module CEECCICS in data set CEE.SCEERUN. Members CEECOPT and CEEWCOPT in CEE.SCEESAMP provide a samples source and job to change installation wide definitions and to re-compile CEEDOPT.

- **Region wide**: The region-wide options, specific for individual CICS regions, are compiled in CSECT CEEROPT. This CSECT is link-edited into a load module of the same name and placed in a data set in the DFHRPL concatenation of the CICS job control language (JCL). Members CEECOPT and CEEWROPT in CEE.SCEESAMP provide a samples source and job to change region-wide definitions and to recompile CEEROPT. Options specified in a CEEROPT override CEEDOPT options.
- **Application level**: User-supplied application program level run-time options can be compiled in a CSECT with name CEEUOPT. This CSECT must be linked with the application program itself. Members CEEUOPT and CEEWUOPT in CEE.SCEESAMP provide a samples source and job to change application level definitions and to re-compile CEEUOPT so that it can be linked with the application program. Options specified in a CEEUOPT, override CEEDOPT, CEEROPT options, or both.

- **C and PL/I programs**: In C and PL/I programs, the source statements through programming language specific statements can provide run-time options.

Language Environment Authorized Program Analysis Report (APAR) PQ38838 introduced a CICS transaction called CLER that enables you to:

- Display the actually active Language Environment run-time options for this CICS region
- Write the run-time options to the CESE TD queue for printing
- Modify a subset of the run-time options

CLER is a conversational transaction. Figure 15-1 shows the initial screen.

![CLER transaction initial screen](image)

**Figure 15-1  CLER transaction initial screen**

### 15.1.3 Monitoring control table

You can define user data fields in performance class monitoring records. User-defined fields are called event monitoring points (EMPs). EMPs allow you to add up to 16K of your own data in each performance monitoring record. The data can consist of a combination of up to 256 counters, 256 clocks, and a single character string of up to 8192 bytes. EMPs have to be specified in the monitoring control table (MCT). The MCT is fully described in the *CICS Transaction Server for z/OS CICS Resource Definition Guide*, SC34-5990.

An EMP is defined with an ID that consists of a NAME and a POINT value. The NAME qualifies the POINT value. EMPs are invoked by the execution of the following command:

```
EXEC CICS MONITOR ENTRYNAME(name) POINT(number)
```
15.1.4 RUWAPOOL SIT option

In the past, when a program ran in an LE environment, a new run unit work area (RUWA) was acquired every time a program issued an EXEC CICS LINK to another program. In CICS Transaction Server (TS) V1.3 and V2.2, the option is available to reuse RUWAs on repeated invocations of applications. This is controlled by the RUWAPOOL SIT parameter.

RUWAPPOL=YES means that CICS creates a pool of storage the first time a program invoked by LE runs in a task. This option provides an available storage pool, which reduces the need to GETMAIN and FREEMAIN RUWAs for every EXEC CICS LINK request.

RUWAPOOL=NO disables the option and provides no RUWA storage pool. Every EXEC CICS LINK to an application results in a GETMAIN for RUWA storage.

Note that the RUWAPOOL parameter affects only application programs running with the LE run-time option ALL31(ON).

15.2 Scenario description

We first compile an MCT containing the required EMPs to show the use of user clocks. We show the sample programs that contain the corresponding monitoring points to start and stop these clocks. A first run shows the CBLPSHPOP overhead when calling a subroutine without any CICS command or other COBOL instruction other than GOBACK. To make it more realistic, we do the same test but with a subroutine containing some EXEC CICS commands. With CICS PA, we produce reports and extracts to analyze the results.

In a similar way, we look at EXEC CICS LINK performance in conjunction with the RUWAPOOL SIT option. We then can also compare EXEC CICS LINK performance with CALL.

The CICS PA reports used afterwards to show the results of the different program runs are performance list reports. To do some manual calculations on the report data, we also use the CICS PA export function to have a data set in tabular form that can be downloaded to a workstation and included in a spreadsheet.

15.3 Measuring CALL performance

In this section, we analyze performance implications of the use of the CALL command.

15.3.1 Sample MCT

Example 15-1 shows the part of the MCT we used for this chapter.

![Example 15-1 Sample MCT](image)
In this example, we use the EMPs defined with ID name EMP1. EMP1.1 defines two clocks, one measuring CPU time and one measuring elapsed time. The names assigned to the clocks are CPUCLOCK and ELAPSCLK. The number 1 in the CLOCK operand is the number assigned to the first defined clock. Number+1 is assigned to subsequent clock names. The PERFORM operand defines the action to be taken to update the user defined fields. Here we use SCPUCLK(1) to start clock 1, the number that was assigned to our clock with CPUCLOCK. Similarly, we use SCLOCK(2) to start the secondly defined clock with name ELAPSCLK.

The CLOCK operand of EMP1.2 references the same clock names. The PERFORM operand, by specifying PCPUCLK and PCLOCK, asks CICS to stop the clocks when an EXEC CICS MONITOR with POINT ID of 2 is executed.

15.3.2 First run: Program description

We developed two small sample COBOL programs. Example 15-2 shows the calling program.

Example 15-2  COBCALL1: Sample COBOL calling program

```
WORKING-STORAGE SECTION.
  01 EMP1 PIC X(8) VALUE 'EMP1'.
  01 LOOP-NUM PIC 99999.
PROCEDURE DIVISION.
BEGIN.
  EXEC CICS MONITOR POINT(1) ENTRYNAME(EMP1)
  END-EXEC.
  MOVE 0 TO LOOP-NUM.
  PERFORM CALL-LOOP UNTIL LOOP-NUM = 100.
  PERFORM DONE.
CALL-LOOP.
  CALL 'COBCALL2'.
  ADD 1 TO LOOP-NUM.
DONE.
  EXEC CICS MONITOR POINT(2) ENTRYNAME(EMP1)
  END-EXEC.
  GOBACK.
```

To avoid the overhead of execution of too many EXEC CICS MONITOR commands, we chose to start and stop the clocks only once and to place them outside of the calling loop.

For the called program, in first instance we chose to have an “empty” program, just performing a GOBACK. Example 15-3 shows this program.
Example 15-3  COBCALL2: ‘Empty’ COBOL called program

PROCEDURE DIVISION.
BEGIN.
  GOBACK.

Attention: You must use extreme care when using EMP clocks. As opposed to CICS clocks, a user clock continues to run when control is passed to the CICS dispatcher. This means that, if you start a CPU clock and do not stop this clock before you run a CICS command that goes through the CICS dispatcher so that a higher priority task can take control, the CPU time executed for that task is accumulated in your CPU clock.

To avoid program loading, we chose to execute a static program call. The main and only program that is known to CICS is COBCALL1. The called subprogram is COBCALL2, and it is linked into the load module COBCALL1. By changing the LOOP-NUM counter in the program, we changed the number of times we call the subprocess. We decided to have three different runs and call the subprocess 100, 5000 and 10000 times. Finally, we define transaction KALL to run the program COBCALL1. As we are using program autoinstall, we did not have to provide PROGRAM definitions.

15.3.3 Executed scenario

The system we were running on is a test system where CICS systems are running with tracing on. In the case of running with CBLPSHPOP OFF, there are no EXEC CICS commands during a COBOL CALL command. With CBLPSHPOP ON, we have these CICS commands and running with a trace on. There is additional overhead compared to the run without these commands. That is why we decided to switch all tracing off for these exercises.

We followed these steps:
1. Run CETR to switch off all tracing.
2. Compile and link the called program COBCALL2.
3. Compile and link the calling program COBCALL1.
4. Execute the CEMT transaction to NEWCOPY program COBCALL1.
5. Execute the CLER transaction to set CBLPSHPOP to ON.
6. Execute the KALL transaction six times.
7. Execute the CLER transaction to set CBLPSHPOP to OFF.
8. Execute the KALL transaction five times.

After this, we recompiled the program to change the number of times to call the subroutine and repeated this scenario.

The first time we execute KALL six times because we expect a program load after the newcopy or because it is the first execution after CICS start.

15.3.4 Performance List report generation

After switching the SMF data sets, we used CICS Performance Analyzer in the following way to select the KALL records. The data set containing the SMF records after the switch is SMFDATA.ALLRECS.G8734V00.

In CICS Performance Analyzer, on the Primary Option Menu, we select 1 to go to the System Definitions screen (Figure 15-2).
On this screen, we entered NEW to create a new System Definition. On the pop-up screen, we entered the new System Name which is the APPLID of the CICS on which we run our tests. In this case, this is SCSCPJA6. For System Type, we chose option 1 to select a CICS System.

We also specified a Dictionary data set name. Because we switch SMF data sets all the time, it can be that the current SMF data set does not contain a dictionary record. In our case, with EMPS defined in the MCT, we could not use the default dictionary record and so we had to define and populate the dictionary data set. The data set itself is created automatically when you specify its name on this screen. To populate it, we placed the cursor under Dictionary in the action bar and pressed Enter. Then we selected 1 as the only possible option to populate a Dictionary data set with the Dictionary record.

We filled in the SMF data set name directly on this screen. We could have added its name to the list of SMF data sets and then selected it from that list. Figure 15-3 shows the final screen after all information is filled in.

We pressed F3 three times to return to the Primary Option Menu. On this screen, we selected option 3 to create a new Report Form. Figure 15-4 shows the screen where we chose to create the new CALL Report Form.
On the pop-up screen, we chose to have a LIST form. In the Edit List Report Form screen, at the end of the list of all possible CMF performance fields, we find the user-defined EMP fields. With the ISPF move line command, we moved CPUCLOCK and ELAPSCCLK before the EOR indicator. We moved also the start time to be included in the report. From the other fields that are before the EOR indicator, we only kept TRAN, PROGRAM, TASKNO, CPU and DISPATCH. Finally, we duplicated the DISPATCH field and on the second field we specified COUNT instead of TIME as the field type.

Figure 15-5 shows the new screen content. Only the fields in front of the EOR indicator appear in the report.
We pressed F3 to save this form. Back on the Primary Option Menu, we chose option 2 to go to the Report Sets screen. On the command line, we entered `NEW CALL` to create a new Report Set with name CALL. Figure 15-6 shows the resulting screen.

![Figure 15-6 Report Set selection screen](image)

In the Performance Reports category, we selected **List**. On the next screen, we entered the APPLID, the form name, and selected the selection criteria Performance as shown in Figure 15-7.

![Figure 15-7 Performance List Report selection criteria](image)
In the resulting screen, we chose to print only the KALL transaction. This was done by entering an I in the Inc/Exc column to indicate that this is an include option. In the Field Name column, we entered TRAN to indicate that the include selection is based on the transaction name and as the first Value, we entered the transaction code KALL. Figure 15-8 shows the resulting screen.

![Image of Performance Select Statement screen]

We then returned to the Report Set screen as shown in Figure 15-6. Now the Global and List options are flagged as active with a Yes in the Active column. On the command line, we entered the RUN command to run the report. This generates the batch job JCL to print the report. On the Run Report Set screen, we typed a slash ('/') in front of Edit JCL before submit to edit the JCL before submitting it. Example 15-4 shows the JCL.

Example 15-4  Generated JCL for batch report

```
//  JOB
/*JOBPARM SYSAFF=SC66
/* CICS PA V1R3 Report JCL
//CICSPA EXEC PGM=CPAMAIN
//STEPLIB DD DSN=CPSA.SCPALINK,DISP=SHR
//SYSPRINT DD SYSOUT=*                  // SMF Input Files
//SMFIN001 DD DSN=SMFDATA.ALLRECS.G8734V00,
//                  DISP=SHR
//* Command Input
//SYSIN DD *
* Report Set =CALL
* Description=CICS PA Report Set
* Reports for System=SCSCPJA6
* Description=Performance testing
CICSPA IN(SMFNO01),
  APPLID(SCSCPJA6),
  LINECNT(60),
  FORMAT('::','/'),
  LIST(OUTPUT(LIST0001),
    SELECT(PERFORMANCE(
      INC(TRAN(KALL)))),
    FIELDS(TRAN,
      PROGRAM,
      TASKNO,
      File  Edit  Object Lists  Options  Help
------------------------------------------------------------------------------------------------------------------------
| CALL - Performance Select Statement | Row 1 of 9 More: > |
Command ===>

<table>
<thead>
<tr>
<th>Active</th>
<th>Report Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc</td>
<td>Start</td>
</tr>
<tr>
<td>Exc</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td>MM/DD/YYYY</td>
</tr>
<tr>
<td></td>
<td>HH:MM:SS.TH</td>
</tr>
</tbody>
</table>

------------------------------------------------------------------------------------------------------------------------

Inc Field --- Value or Range --- Object
/ Exc Name + Type Value/from To List +
INC TRAN KALL
```

Figure 15-8  Performance Select Statement screen
We submitted the job and received the output listing shown in Figure 15-9.

<table>
<thead>
<tr>
<th>Tran Program</th>
<th>TaskNo</th>
<th>Start Time</th>
<th>User CPU Time</th>
<th>Dispatch Time</th>
<th>Dispatch Count</th>
<th>CPU Clock Time</th>
<th>Elapsed Clock Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>KALL COBCALL1</td>
<td>486</td>
<td>19:23:40.397</td>
<td>.0006</td>
<td>.0008</td>
<td>1</td>
<td>.0003</td>
<td>.0003</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>487</td>
<td>19:23:41.357</td>
<td>.0006</td>
<td>.0009</td>
<td>1</td>
<td>.0003</td>
<td>.0004</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>488</td>
<td>19:23:42.319</td>
<td>.0006</td>
<td>.0009</td>
<td>1</td>
<td>.0003</td>
<td>.0004</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>489</td>
<td>19:23:43.394</td>
<td>.0006</td>
<td>.0012</td>
<td>1</td>
<td>.0003</td>
<td>.0003</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>490</td>
<td>19:23:44.506</td>
<td>.0006</td>
<td>.0008</td>
<td>1</td>
<td>.0003</td>
<td>.0003</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>491</td>
<td>19:24:10.170</td>
<td>.0003</td>
<td>.0005</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>492</td>
<td>19:24:10.170</td>
<td>.0003</td>
<td>.0005</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>493</td>
<td>19:24:11.061</td>
<td>.0003</td>
<td>.0043</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>494</td>
<td>19:24:11.899</td>
<td>.0003</td>
<td>.0005</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>495</td>
<td>19:24:12.821</td>
<td>.0003</td>
<td>.0005</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>496</td>
<td>19:24:13.704</td>
<td>.0003</td>
<td>.0005</td>
<td>1</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>499</td>
<td>19:25:41.526</td>
<td>.0156</td>
<td>.0215</td>
<td>3</td>
<td>.0148</td>
<td>.0166</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>500</td>
<td>19:25:42.371</td>
<td>.0152</td>
<td>.0173</td>
<td>1</td>
<td>.0149</td>
<td>.0169</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>501</td>
<td>19:25:43.218</td>
<td>.0151</td>
<td>.0166</td>
<td>1</td>
<td>.0148</td>
<td>.0162</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>502</td>
<td>19:25:44.097</td>
<td>.0152</td>
<td>.0175</td>
<td>1</td>
<td>.0149</td>
<td>.0169</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>503</td>
<td>19:25:44.987</td>
<td>.0152</td>
<td>.0167</td>
<td>1</td>
<td>.0149</td>
<td>.0168</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>504</td>
<td>19:25:45.907</td>
<td>.0153</td>
<td>.0197</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>505</td>
<td>19:25:46.895</td>
<td>.0162</td>
<td>.0181</td>
<td>1</td>
<td>.0148</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>506</td>
<td>19:25:47.705</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>507</td>
<td>19:25:48.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>508</td>
<td>19:25:49.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>509</td>
<td>19:25:49.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>510</td>
<td>19:25:50.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>511</td>
<td>19:25:51.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>512</td>
<td>19:25:52.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>513</td>
<td>19:25:53.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>514</td>
<td>19:25:54.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>515</td>
<td>19:25:55.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>516</td>
<td>19:25:56.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>517</td>
<td>19:25:57.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
<tr>
<td>KALL COBCALL1</td>
<td>518</td>
<td>19:25:58.549</td>
<td>.0171</td>
<td>.0200</td>
<td>1</td>
<td>.0149</td>
<td>.0191</td>
</tr>
</tbody>
</table>

We see that transactions with numbers 485, 499, and 514 have a dispatch count of 3 because they were the first transactions executed after a NEWCOPY of the program. All other runs were dispatched only once so that we are sure that CPU and elapsed wait time only apply to these particular runs.

Transactions 486 to 496 represent the program runs where the subprogram was called 100 times, 500 to 511 are for the runs where the subprogram was called 5000 times, and 515 to 525 are for the runs of 10000 calls.
As expected, the CICS measured transaction CPU time is slightly higher than the EMP CPU clock time. In case of larger applications where smaller isolated parts are measured, this difference may be higher. The same applies for the CICS measured dispatch time and the EMP measured elapsed time, except for the runs where the dispatch count is 3. Here we see a higher dispatch time because it includes the time being dispatched to load the program before the start of the program execution.

For the runs when we call the subprogram 100 times, we see CPU and elapsed times as being zero. The execution time was too short to have it in the fourth digit after the comma.

### 15.3.5 Data export

With the intention of performing some calculations on the CPU measurement results, we planned to export only the two CPU clock fields to a data set. To have only two columns with those two fields, we created a new Report Form called CALL2. On the Report Forms screen, we entered `NEW CALL2` and received the New Report Form pop-up screen. Here we chose to create CALL2 as a model from CALL as shown in Figure 15-10.

![Figure 15-10 New Report Form screen](image)

In the CALL2 form, we moved the fields CPU and CPUCLOCK to the top of the list and moved the EOX indicator behind them as shown in Figure 15-11.
In the CALL Report Set, this time we deactivated the list report option by entering a D in front of List and then selected the Export option in the Extracts category. On the screen shown in Figure 15-12, we entered the APPLID, SCSCPJA6; the Data Set Name to contain the extract, CICSLS2.CALL.EXPORT; specified 1 for the data set disposition; entered CALL2 for the Extract Format Form; and finally, entered S to select the Performance selection criteria.

**Figure 15-11** CALL2 Report Form screen

**Figure 15-12** CALL - Export screen
On the Performance Select Statement screen (Figure 15-13), we entered two selection criteria. One is in the TRAN field to only list the KALL transactions. The other one is in the DISPATCH COUNT field to only list the transactions having a dispatch count of 1. This eliminates the transaction runs where we had a program load.

![Figure 15-13 Export selection criteria](image)

Back on the first Report Set screen, we re-ran the report and submitted the newly generated JCL. The data set was created. The two columns that we found in the data set were used in a Lotus 1-2-3 spreadsheet. Some calculations were performed shown in Table 15-1.

This spreadsheet shows the average total CPU and average CPUCLOCK times calculated per group of execution runs. The Program Execution Ratio column shows the total CPU time for a run with CBLPSHPOP divided by the corresponding total CPU time for a run without CBLPSHPOP. The SUM CALL+PUSH+POP column is the difference between the values of Average CPUCLOCK measured in two runs: with CBLPSHPOP ON and CBLPSHPOP OFF. It gives the CPU time consumed by LE to call the subroutine including the EXEC CICS PUSH HANDLE and the EXEC CICS POP HANDLE command. The last column shows this value divided by the number of calls done to the subroutine and thus is the cost of one call of the subroutine.

Because of the few number of transaction runs, the cost of one call to a subroutine seems to be comparable. This cannot be said of the execution ratio. For the 100-run, we presume that the CPUCLOCK time is so small, even zero in this report, that little deviations can result in significant differences in the calculated ratio. For the other two runs, the ratio is high because of the empty subroutine so that the PUSH and POP commands can be considered to be pure overhead.
15.3.6 Second run

To obtain a more realistic result, we decided to add some other EXEC CICS commands to the subroutine. We added five EXEC CICS commands that are normally not causing a pass through the CICS dispatcher. Example 15-5 shows the subroutine that we created.

Example 15-5  Subroutine with EXEC CICS commands

```
WORKING-STORAGE SECTION.
  01  EIBADDR     POINTER.
  01  USERNAME    PIC X(08) VALUE SPACES.
  01  CICS-TIME   PIC X(08) VALUE SPACES.
  01  PGMTIME.
  03  CURRENT-DATE PIC X(10).
  03  FILLER      PIC X(01) VALUE SPACES.
  03  CURRENT-TIME PIC X(08).
  01  AREAPTR     POINTER.
PROCEDURE DIVISION.
BEGIN.
  EXEC CICS ADDRESS EIB(EIBADDR)
  END-EXEC.
  EXEC CICS ASSIGN USERID(USERNAME)
  END-EXEC.
  EXEC CICS ASKTIME
```
We did the same exercise again. After the KALL transactions ran, we switched the SMF data sets. We updated the system definition to contain the new SMF data set name. For the rest, we could reuse the CICS PA definitions to print the new results as shown in Figure 15-14.

The results were again copied to an identical spreadsheet as the one we used for the first report. It is shown in Table 15-2.
Table 15-2  Lotus 1-2-3 spreadsheet created from CICS PA export: Second run

<table>
<thead>
<tr>
<th>Calls/ CBLPSHPOP</th>
<th>Total CPU</th>
<th>CPUCLOCK</th>
<th>AVG Total CPU</th>
<th>Program execution ratio</th>
<th>Avg CPUCLOCK</th>
<th>SUM CALL+PUSH+POP</th>
<th>1 CALL+PUSH+POP</th>
</tr>
</thead>
<tbody>
<tr>
<td>100/ON</td>
<td>0.0021</td>
<td>0.0018</td>
<td>0.0021</td>
<td>0.0018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100/OFF</td>
<td>0.0018</td>
<td>0.0015</td>
<td>0.0018</td>
<td>0.0015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000/ON</td>
<td>0.0882</td>
<td>0.0861</td>
<td>0.0879</td>
<td>0.0871</td>
<td>0.0876</td>
<td>0.08834</td>
<td>0.08624</td>
</tr>
<tr>
<td>5000/OFF</td>
<td>0.0743</td>
<td>0.072</td>
<td>0.0745</td>
<td>0.0722</td>
<td>0.0742</td>
<td>0.07192</td>
<td>0.07192</td>
</tr>
<tr>
<td>10000/ON</td>
<td>0.1761</td>
<td>0.1722</td>
<td>0.1782</td>
<td>0.174</td>
<td>0.1767</td>
<td>0.17664</td>
<td>0.17244</td>
</tr>
<tr>
<td>10000/OFF</td>
<td>0.1477</td>
<td>0.1433</td>
<td>0.1481</td>
<td>0.1438</td>
<td>0.1478</td>
<td>0.1437</td>
<td>0.1437</td>
</tr>
</tbody>
</table>

The calculated value for the time of executing one call is in the same order as the first calculation, only slightly higher. The program execution ratio is now more consistent.

The conclusion from our tests is that it is not obvious how much savings you can obtain by setting the CBLPSHPOP parameter to OFF. The overhead of the two additional CICS commands has to be compared with the rest of the application. However, this example shows that when running modular applications that do a lot of calls to relatively small subroutines, the CPU consumption gain can be considerable by switching off CBLPSHPOP, if the application allows you to do so.

15.4 Measuring the EXEC CICS LINK command performance

In this section, we analyze the performance of the EXEC CICS LINK command.

15.4.1 Program description

For the EXEC CICS LINK performance exercise, we used the same programs as for the CALL. However, this time we replaced the CALL by an EXEC CICS LINK and the GOBACK by an EXEC CICS RETURN. Example 15-6 shows the source of the calling program.
Example 15-6  COBLINK1 - calling program

WORKING-STORAGE SECTION.
  01 EMP1P    PIC X(8) VALUE 'EMP1'.
  01 LOOP-NUM PIC 99999.
PROCEDURE DIVISION.
BEGIN.
  EXEC CICS MONITOR POINT(1) ENTRYNAME(EMP1P)
    END-EXEC.
  MOVE 0 TO LOOP-NUM.
  PERFORM CALL-LOOP UNTIL LOOP-NUM = 10000.
  PERFORM DONE.
CALL-LOOP.
  EXEC CICS LINK PROGRAM('COBLINK2')
    END-EXEC.
  ADD 1 TO LOOP-NUM.
DONE.
  EXEC CICS MONITOR POINT(2) ENTRYNAME(EMP1P)
    END-EXEC.
  EXEC CICS RETURN
    END-EXEC.

Example 15-7 shows the source of the called program.

Example 15-7  COBLINK2: Called program

PROCEDURE DIVISION.
BEGIN.
  EXEC CICS RETURN
    END-EXEC.

Unlike in the previous example, here we had no choice between static or dynamic options. We had two different load modules and thus the risk exists that we go through the CICS dispatcher during the execution of the EXEC CICS LINK command. However, we expected that during different transaction runs that use the same program load modules, these load modules would remain loaded in CICS storage. Again, we can checked in the SMF information whether we were dispatched multiple times. To run these application programs, we defined the transaction LINK. Again, no program definitions were required because we used the autoinstall program.

15.4.2 Executed scenario

We followed these steps:

1. Run transaction CETR to switch off all tracing.
2. Compile and link the calling program COBLINK1 with a loop count of 100.
3. Compile and link the called program COBLINK2.
4. Stop and restart CICS to run with SIT parameter RUWAPOOL=NO.
5. Execute the LINK transaction six times.
6. Recompile the program COBLINK1 with a loop count of 5000.
7. Execute the CEMT transaction to NEWCOPY program COBLINK1.
8. Execute the LINK transaction six times.
9. Recompile the program COBLINK1 with a loop count of 10000.
10. Execute the CEMT transaction to NEWCOPY program COBLINK1.
11. Execute the LINK transaction six times.
12. Recompile the program COBLINK1 with a loop count of 100.
13. Stop and restart CICS to run with SIT parameter RUWAP00L=YES.
14. Execute the LINK transaction six times.
15. Recompile the program COBLINK1 with a loop count of 5000.
16. Execute the CEMT transaction to NEWCOPY program COBLINK1.
17. Execute the LINK transaction six times.
18. Recompile the program COBLINK1 with a loop count of 10000.
19. Execute the LINK transaction six times.

### 15.4.3 LINK transaction characteristics

Before concentrating on the CPU and elapsed timer values, we were interested in seeing the typical characteristics of this LINK transaction. In CICS PA, we could continue to use the existing system definition. We needed to change the SMF data set name.

For the typical characteristics, we decided to create a new Report Form. For that, we selected **Report Forms** on the Primary Options Menu. On the Report Forms screen, we entered the NEW LINK command to create the new form. A pop-up screen appeared on which we entered option 1 to request a list form. On the LIST Report Form screen, we moved program and storage related SMF field names before the EOR indicator to include these in the list report. Figure 15-15 shows the LINK form as we created it.

**Figure 15-15  LINK Report Form**

We saved this form. Then we returned to the Primary Option Menu, where we selected option 2 to create a new Report Set which we called LINK. The choices of the Performance List report and the Report Set selection criteria are identical to those in Figure 15-6 on page 327 through Figure 15-8 on page 328. The difference is that this time, we selected on transaction
The first transaction in this list has a PCLOAD wait count of 3. This transaction was the first user transaction initiated from a screen after the CICS restart. CICS required first to load the program autoinstall URM before the loads of COBLINK1 and COBLINK2. Because of the three program loads, we also had a change mode count of six, since each load requires a switch from QR mode to RO mode for the loading of the program and then back to QR. This also explains the high dispatch count: one initial dispatch and one after each change mode.

We see a PCLINK count of 101 because the initial link to program COBLINK1 is included in this link count together with the 100 EXEC CICS LINK commands to program COBLINK2. The link to URMs is not included in the total program link count but is reported in a separate field, PCLNKURM. For this first transaction, we linked twice to the autoinstall URM, once for COBLINK1 and once for COBLINK2. The first transaction has a higher GETMAIN count in the user, shared storage above the 16Mb line than the following transactions, again because of first initialization, or both.

The transactions with task number 84 and 91 in the top half of the list are the first ones we executed after we relinked COBLINK1 and performed a CEMT NEWCOPY of only COBLINK1.
Transactions with task number 77 to 96 in the top half of the list were running during the first CICS run. For these transactions, we see that the SC31UGet column corresponds to the values in the PCLINK column. It clearly shows that we were running with RUWAPOOL=NO and that a GETMAIN was performed for each link.

After we stopped and restarted CICS, we again started with a lower task number. The transaction with task number 66 here also shows the higher values for dispatch count and change mode count. Also the SC31UGet value is still high and in the order of the number of times we issued the EXEC CICS LINK command. The reason is that if you specify RUWAPOOL=YES, the first run of a transaction is the same as with RUWAPOOL=NO, but CICS keeps a history of the total storage of RUWAs requested to run the transaction. This means that when the transaction is run again, CICS issues a single GETMAIN for a RUWAPOOL for the total amount of storage required. As in our case, if the transaction follows the same code path, CICS allocates the storage from the RUWAPOOL and no further GETMAIN has to be performed. Therefore, we see that for all subsequent runs of the LINK transaction we only have one GETMAIN left.

15.4.4 LINK performance

The same way as we did for the CALL performance, we exported the CPU and CPUCLOCK values to a PDS to be used as input for a Lotus 1-2-3 spreadsheet. We could reuse the CALL2 Report Form that was built in 15.3.5, “Data export” on page 330. See Figure 15-10 on page 330 and Figure 15-11 on page 331.

We also reused the LINK Report Set that we earlier defined for this test. However, this time, on the Report Sets screen, we selected the LINK Report Set for updating. On the Report Set screen, we selected Export in the Performance Extracts category to specify the new data set name CICSL22.LINK.EXPORT, the CALL2 form, and a selection criteria to only print information about the LINK transaction runs that have a dispatch count equal to 1. This is similar to the actions shown in Figure 15-12 on page 331 and Figure 15-13 on page 332.

On the Report Set screen, we also selected List in the Performance Reports category and provided identical information for form and selection criteria. We saw that both List and Export options are flagged as active wit a Yes in the Active column so that both reports are handled in one report run.

We submitted the generated job. We expect an output in the JES spool and an output in the specified data set. Figure 15-17 shows the beginning of the resulting output in the spool. However, the data set will only contain the first two columns because the EOX indicator was moved behind the CPUCLOCK field as shown in Figure 15-11 on page 331.

<table>
<thead>
<tr>
<th>User CPU</th>
<th>CPUCLOCK</th>
<th>Tran Program</th>
<th>TaskNo</th>
<th>Start Time</th>
<th>Dispatch Count</th>
<th>Dispatch Time</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0025</td>
<td>0.0022</td>
<td>LINK COBLINK1</td>
<td>78</td>
<td>21:38:13.465</td>
<td>1</td>
<td>0.0029</td>
<td>21:38:13.486</td>
</tr>
<tr>
<td>0.0025</td>
<td>0.0022</td>
<td>LINK COBLINK1</td>
<td>79</td>
<td>21:38:14.310</td>
<td>1</td>
<td>0.0028</td>
<td>21:38:14.338</td>
</tr>
<tr>
<td>0.0025</td>
<td>0.0022</td>
<td>LINK COBLINK1</td>
<td>80</td>
<td>21:38:15.158</td>
<td>1</td>
<td>0.0163</td>
<td>21:38:16.191</td>
</tr>
<tr>
<td>0.0026</td>
<td>0.0023</td>
<td>LINK COBLINK1</td>
<td>81</td>
<td>21:38:16.003</td>
<td>1</td>
<td>0.0032</td>
<td>21:38:16.035</td>
</tr>
<tr>
<td>0.0025</td>
<td>0.0022</td>
<td>LINK COBLINK1</td>
<td>82</td>
<td>21:38:16.771</td>
<td>1</td>
<td>0.0029</td>
<td>21:38:16.790</td>
</tr>
</tbody>
</table>

*Figure 15-17*  List output with form EMP2
The figures from the created data set were again transferred to a spreadsheet. Similar calculations were made as we did for the CALL performance test. The spreadsheet is shown in Table 15-3.

Table 15-3  LINK test results: First run

<table>
<thead>
<tr>
<th>LINKs/ RUWAPOOL</th>
<th>Total CPU</th>
<th>CPUCLOCK AVG</th>
<th>Total CPU</th>
<th>Program execution ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>100/NO</td>
<td>0.0025</td>
<td>0.0022</td>
<td>0.0025</td>
<td>0.0025</td>
</tr>
<tr>
<td>100/YES</td>
<td>0.0022</td>
<td>0.0019</td>
<td>0.0023</td>
<td>0.0022</td>
</tr>
<tr>
<td>5000/NO</td>
<td>0.1098</td>
<td>0.1095</td>
<td>0.1093</td>
<td>0.1096</td>
</tr>
<tr>
<td>5000/YES</td>
<td>0.0938</td>
<td>0.0934</td>
<td>0.0946</td>
<td>0.0945</td>
</tr>
<tr>
<td>10000/NO</td>
<td>0.2196</td>
<td>0.2193</td>
<td>0.2193</td>
<td>0.2193</td>
</tr>
<tr>
<td>10000/YES</td>
<td>0.1902</td>
<td>0.1899</td>
<td>0.1898</td>
<td>0.1895</td>
</tr>
</tbody>
</table>

This spreadsheet again shows that the program ratio, for the run where the program linked only a hundred times, is inconsistent with the two other runs. As the overhead of one GETMAIN is low compared to the rest of the code path, the calculations of the cost of one GETMAIN did not give accurate results, so we preferred not to include these calculations in this spreadsheet.

We thought it could be of more interest to show a comparison between the program execution with CALL and GOBACK and the program execution with EXEC CICS LINK and EXEC CICS RETURN. To do so, we first made a second run comparable to the second run of the first part of this chapter. As linked to program, we were using the same subroutine containing the five additional EXEC CICS commands. We only had to change the GOBACK to an EXEC CICS RETURN.

Table 15-4 shows the results of this run in a spreadsheet.
Table 15-4  LINK results: Second run

<table>
<thead>
<tr>
<th>LINKS/ RUWAPOOL</th>
<th>Total CPU</th>
<th>CPUCLOCK</th>
<th>AVG Total CPU</th>
<th>Program execution ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>100/NO</td>
<td>0.004</td>
<td>0.0037</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0038</td>
<td>0.0035</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0039</td>
<td>0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0038</td>
<td>0.0035</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0039</td>
<td>0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0037</td>
<td>0.0035</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0037</td>
<td>0.0034</td>
<td>0.0038</td>
<td>1.02105263</td>
</tr>
<tr>
<td>100/YES</td>
<td>0.0038</td>
<td>0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0039</td>
<td>0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0039</td>
<td>0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0037</td>
<td>0.0035</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0037</td>
<td>0.0034</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000/NO</td>
<td>0.1813</td>
<td>0.1788</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.18</td>
<td>0.1777</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.18</td>
<td>0.1777</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1786</td>
<td>0.1764</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1789</td>
<td>0.1766</td>
<td>0.1796</td>
<td></td>
</tr>
<tr>
<td>5000/YES</td>
<td>0.1733</td>
<td>0.1712</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1736</td>
<td>0.1712</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1737</td>
<td>0.1716</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1705</td>
<td>0.1684</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1721</td>
<td>0.17</td>
<td>0.17264</td>
<td>1.04124189</td>
</tr>
<tr>
<td>10000/NO</td>
<td>0.3565</td>
<td>0.3522</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3603</td>
<td>0.3557</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3591</td>
<td>0.3545</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3635</td>
<td>0.3587</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3582</td>
<td>0.3536</td>
<td>0.35952</td>
<td></td>
</tr>
<tr>
<td>10000/YES</td>
<td>0.3459</td>
<td>0.3414</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3461</td>
<td>0.3417</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3441</td>
<td>0.3396</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3445</td>
<td>0.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3456</td>
<td>0.3408</td>
<td>0.34524</td>
<td>1.04136253</td>
</tr>
</tbody>
</table>

Table 15-5 shows the result of comparing the runs with a COBOL CALL statement versus the runs with EXEC CICS commands.

Table 15-5  CALL versus LINK comparison

<table>
<thead>
<tr>
<th>CALL</th>
<th>LINK</th>
<th>CBLPSHPOP</th>
<th>CBLPSHPOP</th>
<th>RUWAPOOL</th>
<th>RUWAPOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBLPSHPOP ON</td>
<td>CBLPSHPOP OFF</td>
<td>RUWAPOOL</td>
<td>RUWAPOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empty subroutine</td>
<td>3.01</td>
<td>0.31</td>
<td>21.89</td>
<td>19.05</td>
<td></td>
</tr>
<tr>
<td>Five commands in subroutine</td>
<td>17.66</td>
<td>14.80</td>
<td>35.95</td>
<td>34.52</td>
<td></td>
</tr>
</tbody>
</table>

The figures in the table are the average total CPU time for one execution of the program expressed in microseconds. They are shown based on the measurements when the subroutine is called 10000 times.

The lowest figure in the table is 0.31 for the CALL to the empty subroutine with CBLPSHPOP OFF. In this case, during the transaction run, only two EXEC CICS MONITOR commands are...
executed, and no other EXEC CICS command is executed during the 10000 times the subroutine is called. With CBLPSHPOP ON, for the same transaction run, one EXEC CICS PUSH and one EXEC CICS POP are executed per CALL, that means 20000 EXEC CICS commands are added. When CALLing the subroutine with the five EXEC CICS commands, we have to add another 50000 EXEC CICS commands to the previous counts so that we come to 50002 for CBLPSHPOP OFF and 70002 for CBLPSHPOP ON. This is reflected in the different results. Note that we did not calculate the time for one EXEC CICS command because each command has a different code path.

For the LINK figures, we have twice the same number of EXEC CICS commands that are executed: 20003 when calling the empty subroutine and 70003 when calling the five commands subroutine. The difference in time is purely the effect of the difference in code path if RUWAPOOL is YES or NO.

When using the subroutine with the five EXEC CICS commands, we can compare the CALL with CBLPSHPOP ON with the LINK results as those runs execute all about 70000 EXEC CICS commands. In this case, we see that there is a remarkable difference between programs using a COBOL CALL statement and EXEC CICS LINK and RETURN commands.

15.5 Conclusion

We conclude from these test runs that the use of CALL instead of EXEC CICS LINK has significant advantage in terms of CPU consumption. If you develop new COBOL applications, consider using CALL commands rather than EXEC CICS LINK commands. In this case, develop subroutines in a way that they do not require EXEC CICS HANDLE commands or that they include their own EXEC CICS PUSH and POP commands so that LE run-time option CBLPSHPOP can be set to OFF.

If you use a lot of EXEC CICS LINK commands, consider defining RUWAPOOL=YES in the SIT if storage utilization allows to do so.

The goal of this topic was to show the ease of use of CICS PA when you want to do your own performance measurement. After you make your setup, it is easy to reuse the previously defined definitions. If by looking at a certain report, a question arises about an SMF field that is not displayed in this report, it is easy to change the Report Form to include the required field. After a new test run, it is often enough to just change the name of the data set that contains the SMF records to produce a new report.
Exception reporting

This chapter provides examples of CICS PA exception reports.
16.1 Exception Class records

Exception data provides information about exceptional conditions suffered by a transaction, such as queuing for a file string, waiting for storage to become available, or waiting for temporary storage. This data highlights possible problems in the CICS system. CICS writes one exception record for each exception condition that occurs.

CICS Performance Analyzer (PA) can produce two reports of exceptions: a List report and a Summary report. To obtain exception class records, you need to ensure that you are recording exception class monitoring records.

The SIT parameter MN is used to specify whether monitoring is on or off at initialization. Use the individual monitoring class parameters to control which monitoring classes are to be active. The SIT parameter MNEXC is used to specify whether exception class monitoring is activated during CICS initialization.

After the CICS region is active, you can view and alter the status of exception class monitoring using the CEMT INQUIRE/SET MONITOR command. CICS writes the exception monitoring class data to SMF data sets.

16.2 Performance List report showing EXWAIT field

For this exercise, we assumed that we were running a performance report and selected just one transaction, IT8. While running the performance reports, we used the CICS Performance Analyzer field EXWAIT to report on the number of exceptions.

To obtain the performance reports, we added the SMF data set to the CICS System Definition in the CICS Performance Analyzer Systems Definition screen. Figure 16-1 shows that we used the data set SMFDATA.ALLRECS.G8633V00 for the CICS APPLID SCSCPAA1.

---

Figure 16-1  Adding the SMF data set to the CICS System Definition
To produce a performance report which includes the EXWAIT field, we created a new Report Form. From the CICS Performance Analyzer main menu, we selected option 3 for the Report Form. We created the Report Form by specifying NEW EXCFORM. See Figure 16-2.

![Figure 16-2 New Report Form EXCFORM](image)

Using the line commands, we moved the EXWAIT field into the Report Form we set up. We also copied it and made the new copy a Type of COUNT. We did this to see how many times we had an exception. See Figure 16-3.

![Figure 16-3 EXCFORM Report Form](image)
To produce a report, we created a new Report Set. From the CICS Performance Analyzer main menu, we selected option 2 for the Report Set. We created a new report called REPIT8 (Figure 16-4).

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2REPS</td>
<td>CICS PA Report Set</td>
<td>2003/10/14</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>DB2REP2</td>
<td>CICS PA Report Set</td>
<td>2003/10/15</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>EJBREP</td>
<td>CICS PA Report Set</td>
<td>2003/10/23</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>EXCREP</td>
<td>CICS PA Report Set</td>
<td>2003/10/23</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>LOGREPS</td>
<td>CICS PA Report Set</td>
<td>2003/10/22</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>LOGREP2</td>
<td>CICS PA Report Set</td>
<td>2003/10/18</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>REPORT1</td>
<td>CICS PA Report Set</td>
<td>2003/10/23</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>REPORT2</td>
<td>CICS PA Report Set</td>
<td>2003/10/25</td>
<td>CICSLS5</td>
</tr>
<tr>
<td>REPORT3</td>
<td>CICS PA Report Set</td>
<td>2003/10/22</td>
<td>CICSLS5</td>
</tr>
</tbody>
</table>
```

Figure 16-4  New Report Set REPIT8

On the Report Set screen, we selected List in the Performance Reports category as shown in Figure 16-5.

```
<table>
<thead>
<tr>
<th>Description . .</th>
<th>CICS PA Report Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
** Reports ** |
- Options       | No                 |
- Global        | No                 |
- Selection Criteria |
  Performance   | No                 |
  Exception     | No                 |
- Performance Reports |
  List          | No                 |
  List Extended | No                 |
  Summary       | No                 |
  Totals        | No                 |
  Wait Analysis | No                 |
  Cross-System Work | No          |
  Transaction Group | No        |
  BTS           | No                 |
```

Figure 16-5  List Performance Report
We added the Report Form we created to the Report Set. We also specified a selection criteria for the transaction IT8. In Figure 16-6, you can see that we specified the Report Form, EXCFORM, and requested selection criteria. We also entered a title of CICS PA Report IT8 for the report.

![Figure 16-6 Performance List Report](image)

As shown in Figure 16-7, we specified the selection criteria we wanted. In this case, we wanted to include the transaction IT8.

![Figure 16-7 Performance Selection](image)
We ran the Report Set, REPIT8. Now, because we did not specify a CICS APPLID in any other place, when we came to the submit, we received the message "System not specified". As shown in Figure 16-8, we specified the CICS APPLID, SCSCPAA1, on which we wanted to run the Report Set.

<table>
<thead>
<tr>
<th>File Systems Options Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Report Set REPIT8</td>
</tr>
<tr>
<td>System not specified</td>
</tr>
<tr>
<td>Command ===&gt;</td>
</tr>
</tbody>
</table>

Specify run Report Set options then press Enter to continue submit.

System Selection:
- CICS APPLID . scscpaa1 + Image . + Group . +

/ Override System Selections specified in Report Set

----- Report Interval -----

Missing SMF Files Option: MM/DD/YYYY HH:MM:SS.TH
1. Issue error message From
2. Leave DSN unresolved in JCL To
3. Disregard offending reports

Enter "/" to select option
/ Edit JCL before submit

Figure 16-8 Specifying the CICS APPLID
Chapter 16. Exception reporting

16.3 Exception List report

To see the exceptions, we used the CICS Performance Analyzer Exception report. In Figure 16-10, we requested a new Report Set by specifying NEW EXCREP.
To produce an exception report, we selected List in the Exception Reports category as shown in Figure 16-11.
We pressed Enter and the Exception List Report screen is displayed. We entered a title for the report and left the other fields as the default (Figure 16-12).

![Figure 16-12 Report title](image)

We completed the changes for the Exception List Report screen. We pressed F3 twice to return to the Report Set screen. On this screen, we entered line action RUN on the List to run the report (Figure 16-13).

![Figure 16-13 Run Exception List Report](image)

We pressed Enter, and then CICS Performance Analyzer presented the Run Report Set screen (Figure 16-14). On this screen, we verified the CICS APPLID on whose exceptions we wanted to report.
Figure 16-14   Specifying CICS APPLID

Figure 16-15 shows the report that was produced. From the performance report, you can see that task number 59774 had an exception. You can also see that the exception was a string wait on file DEPSUMDB.
When we set up the Report Set for the exceptions, we did not specify any selection criteria. This resulted in the report in Figure 16-15, which also shows that the APPLID SCSCPAA1 suffered exceptions of Waits for EUDSA.

### 16.4 Exception Summary report

We ran the Exception Summary report (Figure 16-16). Here we first deactivated the Exception Report: List and then activated the Exception Report: Summary. We could have run both the List and Summary reports together in the same job, by activating both reports on the Report Set screen.

```
** Reports **  Active
- Options      No
  Global       No
- Selection Criteria No
  Performance  No
  Exception    No
+ Performance Reports No
- Exception Reports No
  List         No
  a Summary    No
- Transaction Resource Usage Reports No
  File Usage Summary No
  Temporary Storage Usage Summary No
  Transaction Resource Usage List No
- Subsystem Reports No
  DB2          No
  WebSphere MQ No
```

**Figure 16-16  Activate Exception Summary Report**

We then ran the Exception Summary report. Figure 16-17 shows the output.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>/FOR</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.046</td>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITB</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.035</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.045</td>
<td>51</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 16-17  Exception Summary report**
The Exception Summary report summarizes the exception records collected by the CICS Monitoring Facility. The records are summarized by transaction identifier. The report provides the total number of exceptions for each transaction. In the report shown in Figure 16-17, you can see that the transaction IT8 had a total of 74 exceptions. Five of them were storage waits and the rest were string waits.

In conclusion, the CICS PA Exception Reports showed the number and type of exceptions from which the CICS System was suffering.
Chapter 17. Analyzing overall system performance

In this chapter, we generate reports that help us understand the overall performance behavior of a system that is composed of a number of CICS regions running a mixed transaction workload. We do not use the detailed step-by-step approach as we do in other chapters. Here we describe which definitions we created and only show the most relevant screens.
17.1 Making a subset of CMF performance records

For this chapter, we were collecting CICS Monitoring Facility (CMF) performance records from nine different systems. Refer to Chapter 5, “System setup and scenario overview” on page 129, for a description of our system setup and CICS region connectivity.

Four of our regions are at CICS TS V1.3 level:

- SCSCPTA1 - TOR
- SCSCPTA2 - TOR
- CICSPAA1 - AOR
- CICSPAA4 - AOR

Four other regions are at CICS TS V2.2 level:

- CICSPLA1 - TOR
- CICSPLA2 - TOR
- CICSPJA6 - AOR
- CICSPJA7 - AOR

We also used a stand-alone CICS Transaction Server (TS) V1.3 region, SCSCPAA6, in which we executed some additional CICS Web-based applications.

Because multiple regions are involved, we started by creating a new system definition. We created an MVS image with the name of SC66. From the time we had the load on our CICS systems, we isolated a 15 minutes period. We started from three system management facility (SMF) data sets that we added to our newly created MVS image system definition as shown in Figure 17-1.

---

We then created a new Report Set with the name CROSS. On the Report Set screen, we selected Record Selection in the Extracts category. Figure 17-2 shows the information that we provided.
After we saved this information, we received a summary screen with the information from Figure 17-2. We repeated this summary three times and filled in the generic CICS APPLIDs to arrive at the screen shown in Figure 17-3.

We ran the report but before we submitted the job. Then we filled in the required time limits of the period that we decided to investigate as shown in Figure 17-4.
We submitted the batch job. CICS Performance Analyzer (PA) created a data set CICSLS2.BIGRUN1 that contains only the required records we are interested in for the rest of our tests. In the system definition of MVS image SC66, we replaced the names of the three SMF data sets with the single data set name CICSLS2.BIGRUN1.

17.2 Working with different CICS system releases

Each new release of CICS usually introduces new CMF performance class fields. Therefore, when you analyze data for CICS systems with mixed releases, some tasks include performance data that is not available for other tasks.

Each CICS system (APPLID) has a CMF Dictionary record that defines which fields are applicable to that system. CICS PA keeps a dictionary record for each CICS APPLID that has CMF performance records in the SMF file. CICS PA uses it to extract field values from the performance records when required.

When using Report Forms to tailor your List, ListX or Summary reports, CICS PA may detect that a required field value is not available, in which case, it is reported as "missing". This typically happens when your Report Form specifies a CMF field for a new release of CICS, and your SMF file includes CMF performance data for CICS systems at an earlier release.

Note: You can specify a monitoring control table (MCT) to exclude fields from the CMF Dictionary. CICS PA treats all missing fields the same, regardless of whether they are excluded by the MCT or only applicable to a higher release of CICS.

When CICS PA reads the input data set, it uses the first dictionary record that is encountered to build the field layout that will be used for the rest of the run. When you request a field that
appears only in one of the releases, two different situations can occur depending on whether the field is present in the first dictionary record.

17.2.1 Dictionary record does not contain requested field

If the first dictionary record encountered in the input data set is from a release that does not contain a requested field, a CPA0311E message is issued and the associated field is omitted. The report that we now are going to produce is a Performance List report in which we use the CICS PA provided sample Report Form CPULST. However, to show the system name of the CMF performance record, we changed the USERID field to APPLID. Figure 17-5 shows that the first dictionary record read was for SCSCPAA4, which is a CICS TS V1.3 system. Message CPA0311E is issued twice because CMF performance fields RO CPU Time and KY8 CPU Time were requested. These correspond to DFHTASK S270 and S263, respectively.

Figure 17-5 System messages from a list report showing CPA0311E

Figure 17-6 shows the beginning of the corresponding Performance List report. The header shows that fields DFHTASK S270 and S263 are missing.

In an attempt to bypass this problem, we made two subsets of CICSL2.BIGRUN1. One subset contains only CICS TS V1.3 records and the other only CICS TS V2.2 records. An...
alternative to creating data sets that contain subsets for specific CICS regions is to create a
group definition. Option 3, Maintain Group definitions, on the Systems Definitions Menu
screen, allows you to do so. A system definition has to exist for an individual system that you
want to add to a group. Figure 17-7 shows the definition of the group we created to contain
only our CICS TS V1.3 regions.

Figure 17-7  CICSTS13 Group definition

The name of this group can then be specified in the Group field of the Record selection
Extracts screen as shown in Figure 17-8. The Image name was removed to contain only the
group name. One of the individual systems from the group has to contain the name of the
SMF input data set so that it can be picked up from there.

Figure 17-8  Record Selection Extracts screen

We changed the SC66 system definition so that it now has two input SMF data sets,
CICLS2.BIGRUN1.CICSTS22 and CICLS2.BIGRUN1.CICSTS13. By defining
CICLS2.BIGRUN1.CICSTS22 before CICLS2.BIGRUN1.CICSTS13, we made it so that
the CICS TS V2.2 dictionary record would be read and used first.

17.2.2 Dictionary record contains requested field

If the first dictionary record encountered is from a release that has the requested field
defined, then the field is included in the extract and the subsequent systems that do not have
the field defined will have Missing substituted for the value. This time CICS PA issues a
message CPA0329E. Figure 17-9 shows the system messages for this case.
### 17.3 Looking at overall system performance

We now review the reports that allow us to have a total view of one or a group of systems. In a specific system or within a CICSPlex, we can also look at groups of transactions or component behavior. Different reports are more appropriate in this case.

#### In this last case, where we have a mix of CICS TS V1.3 and CICS TS V2.2 records, if we asked for a Summary report, we would have found Missing everywhere for the RO CPU Time and KY8 CPU Time. The reason is that the average was calculated on reduced task count or on total task count, running separate reports is more appropriate in this case.
17.3.1 Performance Totals report

The Performance Totals report summarizes the total system behavior as reflected in CMF performance records. It gives a system-wide overview of system performance and resource usage. It can be used in a daily follow-up of system behavior and, in a way, to compare regions to each other. We started with a totals report of all our systems. We continued to use our CROSS Report Set definition. On the EDIT Report Set screen, we selected **Totals** in the Performance Reports category. We verified that the System Selection Image field was set to SC66 and did not change the selection criteria. We submitted the batch job, which resulted in seeing page 1 of the output (Figure 17-11). It shows the overall CICS system usage.

![Default Performance Totals report: Page 1](image)

The **Total Elapsed Run Time** is the time calculated by subtracting the start time from the first encountered SMF record from the stop time of the last SMF record. As we isolated a time period of 15 minutes, the 46 minutes that we see here seem to be rather a high value.

On the second page of the report, we receive a more detailed view of the CPU and dispatch statistics. Figure 17-12 shows the beginning of page 2.
In this report, we see a maximum suspend and response time of 1887 seconds. This is a typical value for the CSOL transaction. CSOL is the TCP/IP listener transaction and is an internal CICS transaction. If you specify TCP=YES in the SIT, this transaction is started at CICS initialization. About every 30 minutes, a CMF performance record is written for this transaction. In this example, where we are interested in only application records, we prefer to exclude these records from the report. In other cases, when you want to consider the total system run, they should not be excluded. That is the reason why these internal transactions are not excluded by default. We suggest a list of long running CICS internal transactions that you can consider excluding from your reports:

- CFQR - RLS quiesce
- CFQS - RLS quiesce
- CSHQ - Scheduler services
- CSNC - IRC connection manager
- CSNE - VTAM node abnormal
- CSOL - TCP/IP listener
- CSSY - CICS internal task
- CSTP - CICS internal task
- CSZI - FEPI

If your CICS system is managed by CICSPlex System Manager (SM), you should also consider excluding:

- COIE - CICSPlex SM hearbeat task
- COI0 - CICSPlex SM CMAS communication task
- CONL - CICSPlex SM MAS initialization and control transaction
- CONM - CICSPlex SM monitoring task
- COHT - CICSPlex SM completed task history recorder

Two more transactions to be considered for exclusion are:

- CSKL - TCP/IP default Sockets listener
- CKTI - MQSeries default listener task initiator
We made an Object List, CICSEXCL, containing all the transactions that were mentioned (Figure 17-13). Figure 17-12 shows the content of the Object List.

![Figure 17-13 Object list to exclude CICS internal transactions](image)

We specified the name of this Object list in the performance criteria of our totals report as shown in Figure 17-14.

![Figure 17-14 Exclusion object list specification](image)
After we saved this update, we submitted the batch job again. Figure 17-15 shows page 1 of the report.

![Figure 17-15 Performance Totals report with CICSEXCL: Page 1](image)

We see that the Total Elapsed Run Time now reflects the time period of our test. The CPU values did not change that much because CSOL is not using the CPU on QR or MS that much.

Figure 17-16 shows page 2 of the report after excluding the internal CICS transactions.

![Figure 17-16 Performance Totals report with CICSEXCL: Page 2](image)

From the total suspend time, we can calculate that in the first report four CSOL transactions were added. Comparing the other values, we see some significant changes in the suspend
time and dispatch wait time. However, these figures represent more the figures that we expected for our application tests.

Figure 17-17 shows part of what follows in the Totals report starting from page 3. It shows the resource utilizations statistics. Each data field from the CMF performance record is summarized in Total, Avg/Task and Max/Task. For clock fields, the count and time components are broken down. For the other fields where there is no time component, only the count values are reported.

![Table](image)

Figure 17-17 Performance Totals report with CICSEXCL: Page 3

### 17.3.2 Performance Summary report

Summary reports where clock and count fields are summarized statistically can also be produced with Performance Summary reports. Any field from a CMF performance record can be included in the Performance Summary report so that this allows you to tailor these reports to your more specific reporting requirements. Summary reports allow you also to look at specific transaction or a set of transactions.
Example 17-1 shows the list of sample Summary Report Forms that are available in your Report Forms data set. You can use these as a starting point for further tailoring. To have the Report form effectively available, you must select it from the pop-up screen that you see after you select Samples in the option bar in the Report Forms screen.

Example 17-1   CICS PA provided Sample Summary Report Forms

<table>
<thead>
<tr>
<th>ABNDSEUM</th>
<th>Transaction Abend Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSRQSUM</td>
<td>CICS BTS Request Activity</td>
</tr>
<tr>
<td>COMMWSUM</td>
<td>Transaction Comms Wait Analysis</td>
</tr>
<tr>
<td>CPUSEXTR</td>
<td>CPU Analysis and Extract</td>
</tr>
<tr>
<td>CPUSUM</td>
<td>Transaction CPU Analysis</td>
</tr>
<tr>
<td>DHSUM</td>
<td>CICS Document Handler Analysis</td>
</tr>
<tr>
<td>ENQSUM</td>
<td>CICS ENQueue/Lock Delay Analysis</td>
</tr>
<tr>
<td>FCSUM</td>
<td>File Request Activity</td>
</tr>
<tr>
<td>FCWTSUM</td>
<td>File Wait Analysis</td>
</tr>
<tr>
<td>FDPSUM</td>
<td>First Dispatch Delay Analysis</td>
</tr>
<tr>
<td>FEPISUM</td>
<td>FEPI Request Activity</td>
</tr>
<tr>
<td>ICSUM</td>
<td>Interval Control Activity</td>
</tr>
<tr>
<td>IMSDBSUM</td>
<td>Transaction DBCTL Usage Analysis</td>
</tr>
<tr>
<td>IMSRQSUM</td>
<td>Transaction DBCTL Req Analysis</td>
</tr>
<tr>
<td>IMSSUM</td>
<td>IMS DBCTL PSB Usage Analysis</td>
</tr>
<tr>
<td>JCSUM</td>
<td>Journaling/Logging Activity</td>
</tr>
<tr>
<td>JVMSUM</td>
<td>Java Virtual Machine Analysis</td>
</tr>
<tr>
<td>PCSUM</td>
<td>Program Request Activity</td>
</tr>
<tr>
<td>PSTORSUM</td>
<td>Program Storage Analysis</td>
</tr>
<tr>
<td>RMIDSUM</td>
<td>CICS RMI Analysis - DB2 Overview</td>
</tr>
<tr>
<td>RMIMSSUM</td>
<td>CICS RMI Analysis - IMS Overview</td>
</tr>
<tr>
<td>RMI0VSUM</td>
<td>CICS RMI Analysis - Overview</td>
</tr>
<tr>
<td>RMISUM1</td>
<td>CICS RMI Analysis - Summary (1)</td>
</tr>
<tr>
<td>RMISUM2</td>
<td>CICS RMI Analysis - Summary (2)</td>
</tr>
<tr>
<td>RTETRSUM</td>
<td>Transaction Routing Analysis (2)</td>
</tr>
<tr>
<td>SOAPSUM</td>
<td>SOAP for CICS Usage - Summary</td>
</tr>
<tr>
<td>SSTORSUM</td>
<td>Shared Storage Analysis</td>
</tr>
<tr>
<td>TCLDLSUM</td>
<td>Tclass Delays by Tranclass Name</td>
</tr>
<tr>
<td>TCPIPSUM</td>
<td>Transactions by TCP/IP Service</td>
</tr>
<tr>
<td>TCP5SUM</td>
<td>CICS Support for TCP/IP Analysis</td>
</tr>
<tr>
<td>TCSUM2</td>
<td>Terminal Control Activity (2)</td>
</tr>
<tr>
<td>TD5SUM</td>
<td>Transient Data Activity</td>
</tr>
<tr>
<td>TRAPLSUM</td>
<td>Transactions by Application Tran</td>
</tr>
<tr>
<td>TRARTSUM</td>
<td>Transaction Routing Analysis (3)</td>
</tr>
<tr>
<td>TRATSUM</td>
<td>Transactions by Applid and TOD</td>
</tr>
<tr>
<td>TRR5SUM</td>
<td>Transaction Routing Analysis (1)</td>
</tr>
<tr>
<td>TRTCLS5</td>
<td>Transactions by Tranclass Name</td>
</tr>
<tr>
<td>TRTESUM</td>
<td>Transaction Usage by Terminal ID</td>
</tr>
<tr>
<td>TRT5SUM</td>
<td>Transactions by Time-of-Day</td>
</tr>
<tr>
<td>TRTR5SUM</td>
<td>Transaction Routing Analysis (4)</td>
</tr>
<tr>
<td>TS5SUM</td>
<td>Temporary Storage Activity</td>
</tr>
<tr>
<td>TSW5SUM</td>
<td>Temporary Storage Wait Analysis</td>
</tr>
<tr>
<td>USTORSUM</td>
<td>User (Task) Storage Analysis</td>
</tr>
<tr>
<td>WEB5SUM</td>
<td>CICS Web Support Analysis</td>
</tr>
</tbody>
</table>
As an example, we start to show a CPU Summary report. To avoid “missing” data, we run the report on the CICSTS22 subset of our CMF performance records. Figure 17-18 shows the contents of the CPUSUM report form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Name + S Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAN A</td>
<td></td>
<td>Transaction identifier</td>
</tr>
<tr>
<td>TASKCNT</td>
<td></td>
<td>Total Task count</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>AVE</td>
<td>Transaction response time</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>MAX</td>
<td>Transaction response time</td>
</tr>
<tr>
<td>DISPATCH</td>
<td>TIME AVE</td>
<td>Dispatch time</td>
</tr>
<tr>
<td>CPU</td>
<td>TIME AVE</td>
<td>CPU time</td>
</tr>
<tr>
<td>SUSPEND</td>
<td>TIME AVE</td>
<td>Suspend time</td>
</tr>
<tr>
<td>QRCPUs</td>
<td>TIME AVE</td>
<td>CICS QR TCB CPU time</td>
</tr>
<tr>
<td>MSCPUs</td>
<td>TIME AVE</td>
<td>CICS TCBs CPU time</td>
</tr>
<tr>
<td>ROCPUs</td>
<td>TIME AVE</td>
<td>CICS RO TCB CPU time</td>
</tr>
<tr>
<td>KY8CPU</td>
<td>TIME AVE</td>
<td>CICS Key 8 TCB CPU time</td>
</tr>
<tr>
<td>J8CPU</td>
<td>TIME AVE</td>
<td>CICS J8 TCB CPU time</td>
</tr>
<tr>
<td>L8CPU</td>
<td>TIME AVE</td>
<td>CICS L8 TCB dispatch time</td>
</tr>
<tr>
<td>S8CPU</td>
<td>TIME AVE</td>
<td>CICS S8 TCB CPU time</td>
</tr>
</tbody>
</table>

Figure 17-18  CPUSUM sample Report Form
The form shows that we selected TRAN and sorted the report in ascending order. Figure 17-19 shows the resulting output.

```
<table>
<thead>
<tr>
<th>Tran</th>
<th>#Tasks</th>
<th>Response Time</th>
<th>Dispatch Time</th>
<th>User CPU</th>
<th>QR CPU</th>
<th>MS CPU</th>
<th>RD CPU</th>
<th>KYB CPU</th>
<th>J8 CPU</th>
<th>L8 CPU</th>
<th>S8 CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDTS</td>
<td>14</td>
<td>.0043</td>
<td>.0171</td>
<td>.0016</td>
<td>.0013</td>
<td>.0028</td>
<td>.0011</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>CIRP</td>
<td>3965</td>
<td>.1561</td>
<td>.1278</td>
<td>.0611</td>
<td>.0283</td>
<td>.0011</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>CIRR</td>
<td>15788</td>
<td>.1093</td>
<td>.1285</td>
<td>.0100</td>
<td>.0002</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>CITS</td>
<td>17</td>
<td>.0028</td>
<td>.0012</td>
<td>.0100</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>CSMI</td>
<td>1320</td>
<td>.0059</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>CSOL</td>
<td>3</td>
<td>1887.33</td>
<td>1887.47</td>
<td>.6293</td>
<td>.4393</td>
<td>.1886</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>DB2N</td>
<td>174</td>
<td>.0083</td>
<td>.0297</td>
<td>.0042</td>
<td>.0008</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>DB2U</td>
<td>636</td>
<td>.0021</td>
<td>.0290</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC1N</td>
<td>305</td>
<td>.1069</td>
<td>.0012</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC1</td>
<td>303</td>
<td>.0069</td>
<td>.0297</td>
<td>.0042</td>
<td>.0008</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC1S</td>
<td>1212</td>
<td>.1093</td>
<td>.2366</td>
<td>.0016</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC13</td>
<td>305</td>
<td>.1004</td>
<td>.2279</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC15</td>
<td>1212</td>
<td>.1097</td>
<td>.2366</td>
<td>.0016</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC16</td>
<td>304</td>
<td>.1666</td>
<td>1.2564</td>
<td>.0756</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JC17</td>
<td>304</td>
<td>.1134</td>
<td>1.9295</td>
<td>.0684</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDBM</td>
<td>303</td>
<td>.3321</td>
<td>2.5468</td>
<td>.3084</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB1</td>
<td>306</td>
<td>.1284</td>
<td>1.9091</td>
<td>.0684</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB3</td>
<td>609</td>
<td>.1160</td>
<td>2.4298</td>
<td>.0342</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB4</td>
<td>303</td>
<td>.2013</td>
<td>3.7688</td>
<td>.0925</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB5</td>
<td>1215</td>
<td>.1266</td>
<td>4.8334</td>
<td>.0465</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB6</td>
<td>304</td>
<td>.2036</td>
<td>10.4893</td>
<td>.1744</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>JDB7</td>
<td>304</td>
<td>.1283</td>
<td>.9939</td>
<td>.0687</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQLM</td>
<td>303</td>
<td>.3421</td>
<td>3.9273</td>
<td>.3195</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQLN</td>
<td>302</td>
<td>.1295</td>
<td>2.4471</td>
<td>.0465</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL1</td>
<td>604</td>
<td>.1270</td>
<td>7.1156</td>
<td>.0474</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL3</td>
<td>303</td>
<td>.1093</td>
<td>2.1465</td>
<td>.0687</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL4</td>
<td>302</td>
<td>.1093</td>
<td>2.1465</td>
<td>.0687</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL5</td>
<td>1209</td>
<td>.1232</td>
<td>2.8840</td>
<td>.0461</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL6</td>
<td>303</td>
<td>.1811</td>
<td>1.1894</td>
<td>.0790</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SQL7</td>
<td>302</td>
<td>.1229</td>
<td>.9066</td>
<td>.0464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
</tbody>
</table>

Figure 17-19  Default CPU Analysis Summary report
Since we have CMF performance records of different systems, we can also ask for the list of transactions per CICS system. To obtain this, we only have to add the APPLID to our Report Form. Figure 17-20 shows that we added APPLID as the first sort field. The sort order is also ascending.

![Figure 17-20 Modified CPUSUM Report Form](image-url)
Figure 17-21 shows the new Summary report. You see that the result shows all transactions executed per APPLID. You receive an additional line that gives the averages per APPLID as well.

```
<table>
<thead>
<tr>
<th>APPLID</th>
<th>Tran</th>
<th>#Tasks</th>
<th>Response Time</th>
<th>Response Time</th>
<th>Dispatch Time</th>
<th>User CPU Time</th>
<th>Suspend Time</th>
<th>QR CPU Time</th>
<th>MS CPU Time</th>
<th>RO CPU Time</th>
<th>KY8 CPU Time</th>
<th>JB CPU Time</th>
<th>LB CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCSCPJA6</td>
<td>CIRP</td>
<td>1806</td>
<td>.1686</td>
<td>10.8778</td>
<td>.1375</td>
<td>.0577</td>
<td>.0311</td>
<td>.0001</td>
<td>.0000</td>
<td>.0000</td>
<td>.0567</td>
<td>.0567</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>CSM1</td>
<td>1320</td>
<td>.0010</td>
<td>.0006</td>
<td>.0009</td>
<td>.0005</td>
<td>.0005</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>JCI1</td>
<td>270</td>
<td>1.845</td>
<td>1.0295</td>
<td>.0335</td>
<td>.0767</td>
<td>.0310</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0757</td>
<td>.0767</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>JCI1</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL5</td>
<td>582</td>
<td>.1342</td>
<td>2.8840</td>
<td>.0999</td>
<td>.0453</td>
<td>.0343</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0444</td>
<td>.0411</td>
<td>.0003</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL6</td>
<td>137</td>
<td>.1972</td>
<td>1.1894</td>
<td>.1626</td>
<td>.0754</td>
<td>.0346</td>
<td>.0010</td>
<td>.0000</td>
<td>.0000</td>
<td>.0744</td>
<td>.0708</td>
<td>.0036</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL5</td>
<td>582</td>
<td>.1342</td>
<td>2.8840</td>
<td>.0999</td>
<td>.0453</td>
<td>.0343</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0444</td>
<td>.0411</td>
<td>.0003</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL6</td>
<td>137</td>
<td>.1972</td>
<td>1.1894</td>
<td>.1626</td>
<td>.0754</td>
<td>.0346</td>
<td>.0010</td>
<td>.0000</td>
<td>.0000</td>
<td>.0744</td>
<td>.0708</td>
<td>.0036</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL5</td>
<td>582</td>
<td>.1342</td>
<td>2.8840</td>
<td>.0999</td>
<td>.0453</td>
<td>.0343</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0444</td>
<td>.0411</td>
<td>.0003</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL6</td>
<td>137</td>
<td>.1972</td>
<td>1.1894</td>
<td>.1626</td>
<td>.0754</td>
<td>.0346</td>
<td>.0010</td>
<td>.0000</td>
<td>.0000</td>
<td>.0744</td>
<td>.0708</td>
<td>.0036</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL7</td>
<td>139</td>
<td>.1373</td>
<td>1.0666</td>
<td>.1007</td>
<td>.0459</td>
<td>.0366</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0450</td>
<td>.0403</td>
<td>.0007</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
<tr>
<td>SCSCPJA6</td>
<td>SQL8</td>
<td>149</td>
<td>3.0177</td>
<td>1.055</td>
<td>.0718</td>
<td>.0245</td>
<td>.0009</td>
<td>.0000</td>
<td>.0000</td>
<td>.0000</td>
<td>.0710</td>
<td>.0710</td>
<td>.0000</td>
</tr>
</tbody>
</table>
```

Figure 17-21  CPU Summary report with APPLID added as sort field
On the Performance Summary Report screen, it is also possible to specify an interval time. To use this time interval option, you need to specify the STOP or START field as a sort field. In this case, CICS PA accumulates the data for each interval in the report period and writes a report line for each. Figure 17-22 shows that we set the Time Interval option to 7:30 minutes to have our originally measured interval divided into two separate intervals. You can also see that the Performance selection criteria is active. To eliminate the influence of long running CICS transactions, we preferred to use the CICSEXCL Object List as well.

Figure 17-22   Performance Summary Report: Time Interval selection
Figure 17-23 shows the resulting report. This report shows the start time of the interval in the first column. You see the same pattern as in Figure 17-21 on page 371. The full dashed line separates the time interval reports.

### 17.3.3 Performance List Extended report

The Performance List Extended report provides similar functionality to the Performance List report except that it sorts the CMF performance records prior to reporting. The sort fields are defined in the Report Form, providing the flexibility to present CMF performance data in your preferred order.

One of the sort fields can specify a limit to restrict the output. For example, you can request sorting by transaction ID (ascending) and response time (descending). Specifying a limit of 20 restricts reporting to the 20 worst performing transactions of each transaction ID. At a glance, this report can highlight poor performing transactions. Then by adjusting the Report Form, you can drill down to determine the reasons for poor performance.

As examples, the sample provided Report Forms contain two LISTX Report Forms, BADCPU and BADRESP. We now look at BADCPU. The content of this Report Form is shown in Figure 17-24. It shows TRAN as a first sort field. Within this selection, we sort by CPU time and BADRESP. We now look at BADCPU. The content of this Report Form is shown in Figure 17-24. It shows TRAN as a first sort field.
The output shows the worst CPU consumers per transaction. Since we saw more than two seconds for a CIRP transaction, we were wondering what would be the worst transactions within the whole system. To produce this information, we edited the BADCPU Report Form and removed the A for the TRAN field so that it is no longer a sort field, we changed the limit.
value from 3 to 10 in the Limit field of the CPU field and changed the USERID field to APPLID. Figure 17-26 shows these modifications.

Figure 17-26  Modified BADCPU Report Form

Figure 17-27 shows the result of running with this Report Form.

V1R3M0  CICS Performance Analyzer
Performance List Extended

Top 10 Worst CPU Times

<table>
<thead>
<tr>
<th>Tran</th>
<th>User</th>
<th>CPU</th>
<th>APPLID</th>
<th>TaskNo</th>
<th>Time</th>
<th>Response</th>
<th>Dispatch</th>
<th>Dispatch</th>
<th>User CPU</th>
<th>Suspend</th>
<th>Suspend</th>
<th>Dispatch</th>
<th>Dispatch</th>
<th>IR Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBS</td>
<td>3.2922</td>
<td>SCSCPJA6</td>
<td>9452</td>
<td>15:01:45.653</td>
<td>4.8343</td>
<td>4.8172</td>
<td>43</td>
<td>3.2922</td>
<td>.0171</td>
<td>43</td>
<td>.0023</td>
<td>42</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>JOBD14</td>
<td>2.8532</td>
<td>SCSCPJA7</td>
<td>9952</td>
<td>14:55:13.922</td>
<td>3.7688</td>
<td>3.7538</td>
<td>49</td>
<td>2.8532</td>
<td>.0149</td>
<td>49</td>
<td>.0005</td>
<td>48</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>SQL3</td>
<td>2.3613</td>
<td>SCSCPJA7</td>
<td>13497</td>
<td>15:01:08.403</td>
<td>7.1156</td>
<td>7.0732</td>
<td>31</td>
<td>2.3613</td>
<td>.0424</td>
<td>31</td>
<td>.0225</td>
<td>30</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>CIRP</td>
<td>2.2373</td>
<td>SCSCPJA7</td>
<td>9148</td>
<td>14:53:58.324</td>
<td>10.4893</td>
<td>10.3844</td>
<td>43</td>
<td>2.2373</td>
<td>.1050</td>
<td>43</td>
<td>.0941</td>
<td>42</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>JOBD6</td>
<td>1.0107</td>
<td>SCSCPJA6</td>
<td>2892</td>
<td>14:50:08.372</td>
<td>10.4893</td>
<td>10.3844</td>
<td>43</td>
<td>1.0107</td>
<td>.1050</td>
<td>43</td>
<td>.0941</td>
<td>42</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>SQLN</td>
<td>.9781</td>
<td>SCSCPJA6</td>
<td>3111</td>
<td>14:50:37.149</td>
<td>3.9273</td>
<td>3.9128</td>
<td>28</td>
<td>.9781</td>
<td>.0144</td>
<td>28</td>
<td>.0050</td>
<td>27</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>JOBDN</td>
<td>.9511</td>
<td>SCSCPJA6</td>
<td>8729</td>
<td>15:00:37.624</td>
<td>2.5468</td>
<td>2.5126</td>
<td>29</td>
<td>.9511</td>
<td>.0342</td>
<td>29</td>
<td>.0193</td>
<td>28</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>JCS</td>
<td>.9131</td>
<td>SCSCPJA6</td>
<td>3464</td>
<td>14:51:24.014</td>
<td>2.3669</td>
<td>2.3597</td>
<td>27</td>
<td>.9131</td>
<td>.0072</td>
<td>27</td>
<td>.0013</td>
<td>26</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>JOBDN</td>
<td>.9096</td>
<td>SCSCPJA6</td>
<td>1023</td>
<td>15:02:43.445</td>
<td>1.6267</td>
<td>1.6077</td>
<td>29</td>
<td>.9096</td>
<td>.0190</td>
<td>29</td>
<td>.0012</td>
<td>28</td>
<td>.0000</td>
<td></td>
</tr>
</tbody>
</table>

Figure 17-27  Top 10 of the worst CPU consumers

17.3.4  Cross-System Work report

Another way to look at the total system is to know where transactions are entering the system, where the resulting application programs run, and which type of requests are executed and where. The Cross-System Work report can be of help in this case because it correlates CMF performance records by network UOW ID. Since the Cross-System Work report only produces lists of transaction and no summaries, we recommend that you run this report for a rather short time period of data. You can use this report to understand a transaction flow, probably in combination with the Transaction Group report.

When you print a Cross-System Work report, CICS PA allows you to choose the kind of UOWs to see. We selected to have only those UOWs printed that have more than one record. The way to do so is to set Processing Options to 1 as shown in Figure 17-28.
Figure 17-28  Processing Options selection

Figure 17-29 shows the two different cases we found in our output listing.

The first line is a CIRP transaction executing the initial program DFJIIRP. APPLID SCSCPJA7 states that this Request Processor task executes in system SCSCPJA7. LUNAME and NETNAME indicate that the request is coming from SCSCPLA1. This indicates that an IIOP request entered SCSCPLA1 where the request receiver task decided to send this request to SCSCPJA7. The MRO session ID on which this transaction request entered SCSCPJA7 is <AC3.

The second line is a CSMI transaction executing in SCSCPJA6. This indicates that a function shipping request came from SCSCPJA7. The Transaction Type of UMD indicates that this mirror transaction is for a Dynamic Program Link request. The program linked to is TRADERBL, as indicated in the PROGRAM field. The Request Type field is AP:F---, which means that under this mirror transaction a file request was executed. If the file request was not executed in a DPLed program but was a real function shipped file request, we would have seen a program name of DFHMIRS and a Request Type of FS:F---.
The second group shows a transaction SX6 that is initiated in system SCSCPTA1 from terminal P000. There is no program name provided because the transaction does not execute in this CICS system. Indeed, the Request Type field indicates that this TOR performed a transaction routing towards system ID PAA4. The second line indicates that the SX6 transaction executes with the same name in SCSCPAA4. The name of the initial program executed is DSWSX6VV.

You can tailor this Cross-System Work report to your needs by creating a LIST or LISTX Report Form. For this report, there is no difference between a LIST or LISTX Report Form as the sort sequences from a LISTX Report Set are ignored in this case.

Figure 17-30 shows the CROSSDET LISTX Report Form that we created to have other output in the Cross-System Work report.

![Figure 17-30 List Extended Report Form](image)

To have the report with the information as requested in the CROSSDET Report Form, we added the report name CROSSDET to the Form field in the Cross-System Work Reports panel and re-ran the report. Figure 17-31 shows that transactions belonging to the same network UOW ID are grouped the same way as before, and only the information printed in the individual records is different.
17.3.5 MVS Workload Activity report

There are two options for the MVS Workload Activity report. You can request a list report which is very similar to the Cross-System Work report because it also uses the network UOW ID to correlate transactions that execute in one single CICS region or multiple CICS regions through transaction routing, function shipping, or distributed transaction routing. Otherwise, you can ask for a Workload Activity Summary report that summarizes response time by WLM service and report classes which can be used to set or verify the goals that were defined to the WLM.

To show reporting on different service classes, we defined one of our AORs, SCSCPJA7, to execute in a different service class (CICSWORK) than all the other regions (CICSDFLT). We also changed the CICSPlex SM workload specification to use the GOAL mode rather than the default QUEUE mode that we used for all other scenarios.

Figure 17-32 shows the Workload Activity Report setup screen.

The List report is selected by entering a slash character next to the List option. Figure 17-33 shows the Workload Manager Activity List report.
Chapter 17. Analyzing overall system performance

The MVS Workload Manager divides the life span of a transaction into two phases: a begin-to-end phase and an execution phase. Applied to the CICS environment, this means that the phase where the TOR receives a transaction and ends it is called the begin-to-end phase. The phase where the transaction moves into an AOR, FOR or elsewhere and is processed is called the execution phase. For a more detailed explanation about Workload Manager state information, refer to OS/390 V2R10.0 MVS Workload Management Services, GC28-1773.

The first two lines of the report show CSOL transactions. The TranType column indicates that these are system transactions. This first CSOL transaction is running in CICS system with APPLID SCSCPLA1. Address space SCSCPLA1 has WLM service class CICSDFLT. The second CSOL transaction is running in SCSCPJA7. This address space has service class CICSWORK. Only SCSCPJA7 has service class CICSWORK. The whole installation has only one report class, WASC. As explained in 17.3.1, “Performance Totals report” on page 362, CSOL is running permanently but about every 30 minutes, a CMF performance record is written. This explains the D in the RT field. This is a record output written by a user event monitoring point DELIVER request. The P column displays the phase information.

For the CSOL transactions, we see BTE, standing for Begin-To-End, indicating that this part of the CSOL transaction execution started and ended in this CICS region without going to another CICS region. The completion field is always blank for BTE phase transactions.

The next line shows a CIRR transaction. The U in the TranType field indicates that this is a user transaction. CIRR is an IIOP request receiver task that starts a request processor task but is not related to it. As request receiver tasks and request processor tasks have different network UOW IDs, they are not grouped together like function shipped mirror tasks or transaction routed tasks. A request receiver is considered to run on its own and thus it is shown as a BTE phase task. The record type is T, indicating that the CMF performance record is written at task termination.

The next transaction in the list is a JDB6 transaction. This transaction is running as a request processor task because it is started by an IIOP request receiver task. For the same reason as for the CIRR transaction, a request processor task appears as a single record. This transaction executes an SQL request that is executed by DB2. This means that only a part of the execution took place in SCSCPJA6. The other part took place in DB2 but we do not have
SMF records from the DB2 region. Because only a part of the execution took place in this region, we see EXE N in the combined Phase and Completion fields.

The following part of the report corresponds to the transaction pairs described in 17.3.4, “Cross-System Work report” on page 375. The first transaction is again a request processor task. However, as we saw before, this task does a Dynamic Program Link to SCSCPJA6 where it performs a file request. Here also, because only a part of the execution took place in each region, we see them flagged as EXE N.

Next two records describe the behavior of transaction SX6. This transaction is attached in TOR SCSCPTA1 but is routed to AOR SCSCPAA4. The transaction starts and terminates in the TOR. That is why this task is flagged as a BTE phase task. The execution phase is done completely in the AOR. This is reflected by the EXE Y.

For tasks with phase EXE, the service and report classes are not provided in the List report. However, when CICS starts an EXE phase task on behalf of an originating task, a WLM token is passed with it to identify the service class. The EXE phase tasks always has the same service class as the originating task. This explains why in this last case, the EXE Y task DB2U that runs on SCSCPJA7, have a service class of CICSDFLT, and not CICSWORK. You can verify this in the WLM Summary report shown in Figure 17-34.

Next to the Workload Activity List report, you can also ask for a Workload Summary report which summarizes response times by WLM service and report classes. The report can be run for BTE phase transactions only but you can also ask to include the EXE Y transactions in the report. The selection of the Summary report and the inclusion of the EXE Y tasks, is done on the Workload Activity Report panel under the heading Reports Required. You can see this in Figure 17-32 on page 378.

Figure 17-34 shows the Workload Manager Activity Summary reports arranged by Service Class and by Report Class.

<table>
<thead>
<tr>
<th>Service Class</th>
<th>APPLID</th>
<th>Phase</th>
<th>#Tasks</th>
<th>Average</th>
<th>Std Dev</th>
<th>90% Peak</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSDFLT</td>
<td>SCSCPAA1 BTE</td>
<td>51</td>
<td>.0377</td>
<td>.1073</td>
<td>.1753</td>
<td>.5600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPAA4 BTE</td>
<td>17</td>
<td>111.043</td>
<td>457.767</td>
<td>697.900</td>
<td>1887.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPJA1 BTE</td>
<td>8816</td>
<td>.3441</td>
<td>20.0989</td>
<td>26.1108</td>
<td>1887.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPLA2 BTE</td>
<td>6954</td>
<td>.4033</td>
<td>22.6318</td>
<td>29.4172</td>
<td>1887.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPTA1 BTE</td>
<td>6624</td>
<td>.0356</td>
<td>.0792</td>
<td>.1371</td>
<td>1.2963</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPTA2 BTE</td>
<td>4680</td>
<td>.0412</td>
<td>.0891</td>
<td>.1555</td>
<td>1.1289</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Total</em></td>
<td></td>
<td>27142</td>
<td>.3005</td>
<td>19.8410</td>
<td>25.7367</td>
<td>1887.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Report Class</th>
<th>APPLID</th>
<th>Phase</th>
<th>#Tasks</th>
<th>Average</th>
<th>Std Dev</th>
<th>90% Peak</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>WASC</td>
<td>SCSCPAA1 BTE</td>
<td>51</td>
<td>.0377</td>
<td>.1073</td>
<td>.1753</td>
<td>.5600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPAA4 BTE</td>
<td>17</td>
<td>111.043</td>
<td>457.767</td>
<td>697.900</td>
<td>1887.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPJA7 BTE</td>
<td>32</td>
<td>58.9871</td>
<td>333.661</td>
<td>486.741</td>
<td>1887.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPJA1 BTE</td>
<td>8816</td>
<td>.3441</td>
<td>20.0989</td>
<td>26.1108</td>
<td>1887.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPJA2 BTE</td>
<td>6954</td>
<td>.4033</td>
<td>22.6318</td>
<td>29.4172</td>
<td>1887.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPTA1 BTE</td>
<td>6624</td>
<td>.0356</td>
<td>.0792</td>
<td>.1371</td>
<td>1.2963</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCSCPTA2 BTE</td>
<td>4680</td>
<td>.0412</td>
<td>.0891</td>
<td>.1555</td>
<td>1.1289</td>
<td></td>
</tr>
</tbody>
</table>
As requested, only BTE phase transactions are considered in this report. We see that service class CICSDFLT covers all regions except SCSCPJA7 which runs in service class CICSWORK.

Since there is only one report class, SCSCPJA7 is grouped with the other regions. In this report, again we see some maximum response times of 1887 seconds. As explained in 17.3.1, “Performance Totals report” on page 362, this is because of the presence of the CSOL transaction. Its long response time has a negative influence on the other values of average, standard deviation and the 90% peak value.

Figure 17-35 shows the Workload Manager Activity Summary report produced with the same SMF input records, but here with the transaction group CICSEXCL specified for selection criteria to exclude all CICS long running transactions. We also chose to include the EXE Y phase transactions.

We see that the report includes now also the EXE Y phase transactions. The values of the number of EXE Y phase transactions is one less for SCSCPA4, SCSCPLA1, SCSCPLA2 and SCSCPJA7 compared to the previous report. These are the CSOL transactions that are not counted now. We now have a better distribution in the different values for the response times.

17.3.6 Using DB2

We followed exactly the instructions in 19.4.5, “Export HDB data sets to DB2” on page 441, to load our system data into a DB2 table that we called ITSO.CICSPATB.
In 17.3.3, “Performance List Extended report” on page 373, we calculated the top 10 of the worst CPU consumers. We used CICS PA and the provided sample LISTX Report Form, BADCPU. Using DB2, we show how to produce a report that shows the worst storage consumers.

To do so, we add the fields SC31CHWM (ECDSA) and SC31UHWM (EUDSA) and sort on this sum that we called TOTAL31, as shown in Example 17-2. We also calculate the sum of SC24CHWM (CDSA) and the SC24UHWM (UDSA). Next to that, we show the individual fields that were used in this calculation.

### Example 17-2  Worst storage consumers query

```sql
SELECT
    TRAN,  
    APPLID,  
    TASKNO,  
    (SC31CHWM + SC31UHWM) AS TOTAL31,  
    (SC24CHWM + SC24UHWM) AS TOTAL24,  
    SC31CHWM,  
    SC31UHWM,  
    SC24CHWM,  
    SC24UHWM,  
    SC31GSHR,  
    SC24GSHR
FROM ITSO.CICSPATB A  
WHERE 10 >= (SELECT COUNT(*)  
    FROM ITSO.CICSPATB B  
    WHERE (A.SC31CHWM + A.SC31UHWM)  
        <= (B.SC31CHWM + B.SC31UHWM)  
)  
ORDER BY TOTAL31 DESC;
```

Figure 17-36 shows the output of this query.

```plaintext
--
TRAN | APPLID | TASKNO | TOTAL31 | TOTAL24 | SC31CHWM | SC31UHWM | SC24CHWM | SC24UHWM | SC31GSHR | SC24GSHR
---+-------+--------+---------+---------+---------+---------+---------+---------+---------+---------
CWBA | SCSCPJA6 | 29847 | 8634288 |  576 | 32784 | 8601504 | 0 | 576 | 0 | 0
CWBA | SCSCPJA6 | 29835 | 8634256 |  0 | 32784 | 8601472 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 27816 | 8634240 |  0 | 32784 | 8601456 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 29956 | 8599248 |  0 | 5744 | 8593504 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 28401 | 8599248 |  0 | 5744 | 8593504 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 28405 | 120880 |  0 | 2400 | 118480 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 27822 | 120880 |  0 | 2400 | 118480 | 0 | 0 | 0 | 0
CWBA | SCSCPJA6 | 27822 | 120880 |  0 | 2400 | 118480 | 0 | 0 | 0 | 0
DSNE610I NUMBER OF ROWS DISPLAYED IS 10
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

DB2 Version 7 introduced new SQL statements that allow you to select a limited number of rows. We used the FETCH FIRST 10 ROWS ONLY statement to create a query similar to the one shown in Figure 17-2. The query and the SPUFI output are shown in Figure 17-37.
Chapter 17. Analyzing overall system performance

The output of this SQL statement shows only CWBA transactions in region SCSCPJA6. The query shown in Figure 17-38 lists 10 different transactions with the worst storage usage from all of our CICS regions.

Figure 17-37  DB2 Version 7 query for CWBA transaction

SELECT
TRAN,
APPLID,
TASKNO,
(SC31CHWM + SC31UHWM) AS TOTAL31,
(SC24CHWM + SC24UHWM) AS TOTAL24,
SC31CHWM,
SC31UHWM,
SC24CHWM,
SC24UHWM,
SC31GSHR,
SC24GSHR
FROM ITSO.CICSPATB
ORDER BY TOTAL31 DESC
FETCH FIRST 10 ROWS ONLY

---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--
TRAN  APPLID         TASKNO      TOTAL31      TOTAL24     SC31CHWM     SC31UHWM     SC24CHWM     SC24UHWM     SC31GSHR     SC24 GSHR
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--
CWBA  SCSCPJA6        29847      8634288          576        32784      8601504            0          576            0            0
CWBA  SCSCPJA6        29835      8634256            0        32784      8601472            0            0            0            0
CWBA  SCSCPJA6        27816      8634240            0        32784      8601456            0            0            0            0
CWBA  SCSCPJA6        28401      8599248            0         5744      8593504            0            0            0            0
CWBA  SCSCPJA6        29956      8599248            0         5744      8593504            0            0            0            0
CWBA  SCSCPJA6        29960       120880            0         2400       118480            0            0            0            0
CWBA  SCSCPJA6        29915       120880            0         2400       118480            0            0            0            0
CWBA  SCSCPJA6        29840       120880            0         2400       118480            0            0            0            0
CWBA  SCSCPJA6        28405       120880            0         2400       118480            0            0            0            0
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--

Figure 17-38  Worst storage consumers DB2 Version 7 query

SELECT
TRAN,
APPLID,
MAX(SC31CHWM + SC31UHWM) AS TOTAL31,
MAX(SC24CHWM + SC24UHWM) AS TOTAL24
FROM ITSO.CICSPATB
GROUP BY TRAN,APPLID
ORDER BY TOTAL31 DESC,TRAN ASC
FETCH FIRST 10 ROWS ONLY ;

---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--
TRAN  APPLID        TOTAL31      TOTAL24
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--
CWBA  SCSCPJA6      8634288          576
CLER  SCSCPJA6      97760          1168
CETR  SCSCPJA6     61632          12384
CIRP  SCSCPJA6     33248            0
CSMI  SCSCPJA6     32320            0
CEMT  SCSCPJA6     26256            0
CIRR  SCSCPJA6     21632            0
CIIR  SCSCPJA6     21632            0
CEED  SCSCPJA6     21088           96
CECI  SCSCPJA6     21072           96
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+------- --+--
Using CICS Performance Analyzer reports for problem determination

This chapter demonstrates how to use CICS Performance Analyzer (PA) reports to diagnose a new response time problem. The various CICS Performance Analyzer reports we use to analyze the problem are discussed in detail.
18.1 CICS VSAM problem determination scenario description

To produce useful and realistic performance data for the problem determination scenario project, we used a traditional 3270 COBOL VSAM application as referred to in other projects earlier in this book. We also added one new transaction, BR1.

The base application uses nine VSAM files. We chose to place the files in a File Owning Region (FOR) to mimic a common customer environment. We used the Teleprocessing Network Simulator (TPNS) to simulate a realistic 3270 workload environment.

The 3270 VSAM application workload consists of four business applications:

- **Hotelres**: A simple 3270 hotel reservation application using two VSAM data sets. Two transactions, HX1 and HX2, are available which drive the application.
- **Inventory**: An inventory tracking application. It mainly uses two transactions to manage the inventory. Transaction IX8 updates the inventory, and transaction IX2 is used to inquire about part locations within the inventory.
- **Specification**: A bill of material management application. It uses two transactions (PX2 and PX3) to inquire on information about part lists.
- **Stock**: A stock control application. Transaction SX2 is used to update any inventory, transaction SX4 allows you to insert new vendors, and transaction SX6 is used to delete parts from the stock. Our new transaction BR1 is part of this application. It performs inquiries on the vendor file.

Figure 18-1 illustrates the VSAM file usage of the entire workload. We do not describe non-VSAM related aspects of the workload, such as program and map design, since they are not relevant to our problem determination scenario.

Refer to Chapter 5, “System setup and scenario overview” on page 129, for a full description of the CICS environment setup that we used to run the workload.

![Figure 18-1  3270 VSAM application workload](image)

### 18.1.1 Workload generation

We used Teleprocessing Network Simulator (TPNS) to generate an appropriate 3270 VSAM workload for our scenario. TPNS is a terminal and network simulation tool. You can find a
brief overview of its capabilities in 6.2.1, “RLS workload generation” on page 141. The only difference between what we used for this scenario and what is described in 6.2.1, “RLS workload generation” on page 141, is the list of transactions and the probability of distribution for our workload. Example 18-1 shows our transaction list and how we specified the probability of distribution for our workload scenario.

---

### Example 18-1  Probability of distribution

<table>
<thead>
<tr>
<th></th>
<th>PATH</th>
<th>PX3,PX2,IX8,IX1,IX2,HX1, SX6,SX2,SX4,TX1,BR1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIST</td>
<td>90,100,80,30,60,30, 100,30,20,40,40</td>
<td></td>
</tr>
</tbody>
</table>

---

### 18.1.2 Performance objectives

The performance objectives and priorities depend very much on user expectations. From our previous experiences with the application and the files in an FOR configuration, we know what average response times we can expect for different transactions. From the point of view of a terminal user, this is the most important characteristic.

Table 18-1 summarizes our performance objectives.

#### Table 18-1  VSAM FOR scenario performance objectives

<table>
<thead>
<tr>
<th>Application</th>
<th>Transaction</th>
<th>Response average</th>
<th>Response maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOTEL</td>
<td>HX1</td>
<td>&lt; 0,02</td>
<td>0,2</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>IX8</td>
<td>&lt;0,05</td>
<td>0,3</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>IX2</td>
<td>&lt;0,03</td>
<td>0,28</td>
</tr>
<tr>
<td>SPECIFIC</td>
<td>PX2,PX3</td>
<td>&lt;0,04</td>
<td>0,46</td>
</tr>
<tr>
<td>STOCK</td>
<td>SX2,SX4,SX6 BR1</td>
<td>&lt;0,06,0,03,0,01</td>
<td>0,8,0,3,0,08</td>
</tr>
</tbody>
</table>

We are ready to start the VSAM workload using TPNS. We gather the necessary baseline performance data and analyze the results using a CICS Performance Analyzer summary report.

### 18.2 Running the scenario

This section describes the tasks that we performed to collect the performance data for our VSAM application. First we used CICS Performance Analyzer functionality to provide an overview of the performance behavior of the application. We performed the following steps:

1. Start TPNS and run the entire application for about 15 minutes. The application runs through region PAA1, which is used as both a TOR and AOR, a setup common in many environments, and PTA1, which was optimized to function as an FOR for this scenario.
2. Stop TPNS.
3. Switch the SMF data sets. Note the name of the archived SMF data set that contains the performance data.


### 18.2.1 Collecting SMF data

During our TPNS run, the SMF data sets filled up and switched a number of times. We kept track of the name of each archive SMF data set produced during the test. After we ran TPNS for about 15 minutes, we stopped TPNS, switched SMF data sets, and obtained the name of the final archived SMF data set. We ran a separate extract on all the archived SMF data sets created during the test to create a single data set that we would use for reporting. The sample job in Example 18-2 was used to do this. This is an optional step that we performed for two reasons.

- The archived SMF data on our test system was kept in generation data group (GDG) data sets. We did not want our test data to roll off and be deleted before we completed our analysis.
- We wanted only a subset of all SMF data produced. The sample job in Example 18-2 only selects SMF record types 110, 101, 116, and 88. Reporting on this smaller extract is faster.

#### Example 18-2  Sample job to extract a subset of SMF records

```plaintext
//SMFPA     JOB ACCNT#,'CICSRS7     ',MSGLEVEL=(1,1),
//    NOTIFY=&SYSUID,REGION=5M
//*********************************************************************
//*                                                                   *
//* This job extracts selected SMF records to a separate file. The    *
//* new file contains SMF records used by CICS Performance Analyzer   *
//*********************************************************************
//EXTRACT EXEC PGM=IFASMFDP
//INDD      DD DISP=SHR,DSN=SMFDATA.CICSRECS.G0164V00
//          DD DISP=SHR,DSN=SMFDATA.CICSRECS.G0165V00
//          DD DISP=SHR,DSN=SMFDATA.CICSRECS.G0166V00
//OUTDD     DD DISP=OLD,DSN=CICSRS7.SMF110.TESTCASE
//SYSUDUMP  DD SYSOUT=*  
//SYSPRINT  DD SYSOUT=*  
//SYSIN     DD *           
//           INDD(INDD,OPTIONS(DUMP))
//           OUTDD(outdd,type(110,101,116,88))
//           START(1006)
//           END(1022)
/*
```

For an explanation about how to use program IFASMFDP, see z/OS V1R4.0 MVS System Management Facilities (SMF), SA22-7630.

### 18.2.2 Updating system definitions

CICS Performance Analyzer can automatically populate your system’s definitions with details extracted from SMF files. To check that we are using the correct SMF data set and that we collected performance data for all the systems, we used the Take-up function first.
To perform the data take-up from our SMF extract, we performed these steps:

1. From the CICS Performance Analyzer Primary Option Menu, select option 1.
2. One the System Definitions menu, select option 4.
3. Update the Data Take-Up from SMF screen with the SMF data set name as shown in Figure 18-2. Press Enter to generate a batch job to extract the details from the SMF data set.

When the job completes, you are prompted to update the system definitions with the results of the batch job. The prompt appears when you arrive at the System Definitions screen the next time.

![Figure 18-2 Data Take-Up from SMF screen](image)

When the generated batch job completed, we verified that we had SMF records for MVS image SC66 as well as for the CICS systems participating in the workload. We also checked that the time interval of our TPNS run was processed using this SMF data set. Figure 18-3 shows the output that was produced by the take-up job.

![Figure 18-3 Take-Up from SMF output](image)

Take-up processing updates the system definitions automatically. From the System Definitions Menu, we selected option 1 to open the Systems Definitions screen. Figure 18-4
shows the System Definition screen after processing Take-Up from SMF. It shows that the CICS systems we used were added automatically during take-up processing.

We typed line action S next to the SC66 Image entry to select it. Figure 18-5 shows the MVS Image screen that is displayed.

We verified that the SMF data set name listed was the one we used in our take-up job. We pressed F3 to return to the System Definitions screen. Out of curiosity, we selected a few more entries from the table by typing line action S, and then pressing F3 repeatedly until we returned to the CICS Performance Analyzer Primary Option Menu.

18.3 Producing the baseline Performance Summary report

This section explains how we produced our baseline Performance Summary Report.
Before we could quantitatively measure a reported response time problem, we needed to know how an application performed in the past. Often times we can detect minor changes in performance of tasks before they become noticeable to users, or before minor performance problems become large financial bills for customers with a charge-back system. To understand our current application, we produced a Performance Summary Report.

### 18.3.1 Creating the Object List

One of our goals in producing the reports for our application is to eliminate entries for non-application related transactions. This allows us to focus purely on our application, and not on other tasks. CICS Performance Analyzer gives us the ability to define a list of those transactions which belong to our application and then produce reports and extracts which contain only on those transactions. To do this, we created an Object List.

The process of creating an Object List is discussed in 2.12, “Maintaining Object Lists” on page 56. For our application, we created an Object List called VSAMFSHP with the 10 transactions shown in Figure 18-6.

![Figure 18-6 Object List VSAMFSHP](#)

We included all transactions for our application and the CICS mirror transaction CSMI. We included CSMI because our application environment uses function shipping. File access to remote files by the application was done under the mirror transaction, not our application transaction.

### 18.3.2 Creating the Report Set

To begin creating our report, we created a Report Set. We selected option 2 from the CICS Performance Analyzer Primary Option Menu. The Report Sets screen is displayed.

On the command line, we typed NEW and pressed Enter. On the New Report Set screen, we entered BASELINE for the name of the Report Set, and then pressed Enter. The Report Set screen shown in Figure 18-7 was displayed.
### Figure 18-7 Report Set BASELINE

<table>
<thead>
<tr>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS PA Report Set</td>
<td>Active</td>
</tr>
</tbody>
</table>

- **Options**
  - Global: No

- **Selection Criteria**
  - Performance: No
  - Exception: No

- **Performance Reports**
  - List: No
  - List Extended: No
  - Summary: No
  - Totals: No
  - Wait Analysis: No
  - Cross-System Work: No
  - Transaction Group: No
  - BTS: No
  - Workload Activity: No

- **Exception Reports**
  - List: No
  - Summary: No

- **Transaction Resource Usage Reports**
  - File Usage Summary: No
  - Temporary Storage Usage Summary: No
  - Transaction Resource Usage List: No

- **Subsystem Reports**
  - DB2: No
  - WebSphere MQ: No

- **System Reports**
  - System Logger: No

- **Performance Graphs**
  - Transaction Rate: No
  - Transaction Response Time: No

- **Extracts**
  - Cross-System Work: No
  - Export: No
  - Record Selection: No

**End of Reports**

### 18.3.3 Creating the Performance Summary report

We typed line action S next to Summary within the Performance Reports category. The Performance Summary Report screen shown in Figure 18-8 was displayed.
We typed line action S next to the Performance selection criteria and pressed Enter. The Performance Select Statement screen in Figure 18-9 was displayed. We wanted to produce a report of our application transactions from a five minute interval in the middle of our TPNS test run. Therefore, we specified a selection condition to include only tasks in our object list VSAMFSHP that started on 09/24/2003 between the time 10:06:29.00 and 10:11:29.00 (Figure 18-9.)
We entered both the time and transaction selection conditions on a single Performance Select Statement screen. Both conditions must be met before a record is selected for inclusion in the report (AND processing.) If we created two separate select statements for each condition, then a transaction would be included in the report if it met either one (OR processing.) We pressed F3 three times to return to the Report Set screen.

18.3.4 Submitting the batch job

To submit our job, we typed the RUN line action next to the Summary Report and pressed Enter. The Run Report Set screen was displayed (Figure 18-10.)

![Figure 18-10 Run Report Set](image)

We leave CICS APPLID blank, and only specify Image SC66. We use the / line action to indicate that we want to edit the JCL before submitting the job. We left CICS APPLID blank, and only specified a CICS System Selection criteria of SC66. We entered a forward slash (/) next to Edit JCL before submit because we wanted the option to change the job name for each job. Keeping track of which job ran which report is easier when the job names are unique. We pressed Enter. Then we were placed into ISPF Edit mode for our batch JCL, shown in Figure 18-11.
Figure 18-11 Performance Summary Report JCL

We provide a meaningful job name

We verify the SMF data set is the one we want to use

We see the report options and selection criteria that we entered on the ISPF screens are turned into report selection control statements
After providing a unique job name, we entered SUBMIT on the command line and pressed Enter. When the job completed, it produced the output shown in Figure 18-12.

The average response times are in line with what we expected from this application in the current environment. Only those transactions that were part of our VSAMFSHP Object are reported, and only those that started during our specified time interval.

If this was a live production application, as part of our normal application monitoring procedures, we would want to produce this report daily and distribute it to our applications management. We would set it up to run through our production job scheduling package, where the input SMF data is updated automatically and the reports are sent to an online viewing facility such as IBM Content Manager OnDemand for iSeries®.

### 18.4 Application changes implemented

Next we simulated our applications development group implementing program changes. Programs and maps were updated and moved to production. The next time our regular Performance Summary report ran after the application changes are made, we saw the report in Figure 18-13.

The average response time for the transactions in our application remained within our expected ranges shown in Table 18-1 on page 387 with the exception of transaction BR1. Comparing the data from our current report in Figure 18-13 with that of our baseline report from Figure 18-12, we see a 571% increase in average response time for transaction BR1. Looking closer at transaction BR1, we also see a 588% increase in average CPU time, 581% increase in average suspend time, 1450% increase in average dispatch wait time, and 752% increase in average IR wait time.
Figure 18-13 Performance Summary Report after application changes

We used Lotus 1-2-3 to create a chart of the average response times for the transactions in our application so we could visually compare the changes (Figure 18-14.)

Figure 18-14 Average response time comparison chart
The chart shows the dramatic increase in average response time for transaction BR1. We decided to look closely at the rest of the averages for transaction BR1 (Figure 18-15.)

![Transaction BR1 averages chart](image)

In our five minute sampling, 858 BR1 transactions ran. If this sample was representative of the entire day, a small increase in CPU utilization per transaction could have a large impact, especially for customers who use charge-back systems.

Taking a proactive approach, we decided to investigate the cause of the increase. Because of the increase in the average suspend time and average IR wait time, we ran the Wait Analysis Report.

### 18.5 Producing the Wait Analysis report

To produce the Wait Analysis report, we selected option 2 from the CICS Performance Analyzer Primary Option Menu. Then we typed line action $ next to the Report Set we created in Figure 18-7 on page 392 to return to the Report Sets screen. We typed line action $ next to the Wait Analysis report and pressed Enter. The Wait Analysis Report screen shown in Figure 18-16 is displayed.

![Baseline vs After application changes](image)

We wanted to see the difference that the application changes made, so we produced two separate reports. One report shows a wait analysis before any application changes were made. The other report shows a wait analysis after the changes were made.

For the baseline report, we entered an appropriate title on the screen shown in Figure 18-16, typed line action $ next to the Performance selection criteria option, and pressed Enter.
Chapter 18. Using CICS Performance Analyzer reports for problem determination

Figure 18-16   Wait Analysis Report screen

The Performance Select Statement screen in Figure 18-17 was displayed.

Figure 18-17   Performance Select Statement screen

We entered the same selection criteria that we used to produce our baseline Performance Summary report in Figure 18-9 on page 393 and then pressed F3 three times to return to the Wait Analysis Reports screen (Figure 18-18.)
We wanted to produce both reports with one job, so we entered line action \texttt{R} next to our first selection criteria and pressed Enter. We changed the output DDnames for each selection criteria to indicate which report is being written to it, and then typed line action \texttt{S} next to the second entry to change the date and time selection criteria.

The Wait Analysis Report screen was displayed for our second selection criteria entry (Figure 18-19). Then we changed the title appropriately, typed line action \texttt{S} next to the Performance selection criteria option, and pressed Enter.

We enter a title and line action \texttt{S}
The Performance selection criteria screen was displayed. We typed line action S next to our criteria, and pressed Enter. The Performance Select Statement screen shown in Figure 18-20 was displayed.

![Figure 18-20   Performance Select Statement screen](image)

After changing the time interval to match the Performance Summary report from Figure 18-13 on page 397, we pressed F3 four times to return to the Report Set screen, and entered line action RUN next to the Wait Analysis report. We entered a / next to Edit JCL before submit and pressed Enter. We entered ISPF edit mode for our batch JCL, as shown in Figure 18-21.

![Figure 18-21   JCL for the Wait Analysis Report job](image)

This SMF data set contains records for our baseline interval and our post-application change interval. It is the data set we created in 18.2.1, “Collecting SMF data” on page 388.
We provided a meaningful job name, entered SUBMIT on the command line, and pressed Enter. When the job completed, we had the two reports shown in Figure 18-22 and Figure 18-23. In the reports that were produced, all transactions from our application were reported. However, we only included transaction BR1 in these examples.

### Figure 18-22  Wait Analysis Report before application changes

| Resource Manager Interface (RMI) suspend time | 0.0000 | 0.0000 | 0 | 0 | 0.0% of Suspend |
| Resource Manager Interface (RMI) elapsed time | 0.0000 | 0.0000 | 0 | 0 | 0.0% of Response |

### Figure 18-23  Wait Analysis Report after application changes

We compared the Dispatch Wait Time in the Summary Data section. This figure increased 1450%. It gives us the elapsed time for which the user task waited for redispach by the CICS dispatcher domain. It is the aggregate of the wait times between each wait event completion and the user task being redispachd by the CICS dispatcher domain.

### The Wait Analysis report has two sections:

- The first section provides a summary of common performance metrics.
- The second section provides a detailed breakdown of suspend time by component.
In the Count Average column, before the application changes were made, each BR1 had an average of 6.0 wait events. After the application changes, each BR1 had an average of 77.8 wait events. Each time a task enters a wait state, it gives up control and must be redispached by CICS after the wait event completes. This count tells us that something in the application is making the task give up control almost 13 times more often than it used to.

Next we looked at the Suspend Detail section. This section details the components of the Suspend Wait Time reported in the Summary Data section. The Suspend Detail includes one report line for every Suspend component clock with a non-zero value. The components are reported in descending wait time order. In both reports, IRIOWTT is the wait time that contributes the most to the total task suspend time. However, comparing the average IRIOWTT time from both reports, we see the 752% increase that was shown in the Performance Summary report we produced earlier (Figure 18-13.)

The Wait Analysis report verified that the MRO link wait time is the main contributor to the increase in the total BR1 suspend wait time. It also tells us that each BR1 transaction is incurring approximately 13 times more MRO link wait events than it used to.

Now we need to determine what is causing that increase in MRO link wait events, so we turn to the Cross-System Work report.

18.6 Producing the Cross-System Work report

This section explains how we used the Cross-System Work report to uncover the details of the increased MRO activity for transaction BR1.

From the CICS Performance Analyzer Primary Option Menu, we selected option 2. Then we typed line action S next to the Report Set we created in Figure 18-7 on page 392 to return to the Report Sets screen. We typed line action S next to the Cross-System Work report and pressed Enter. The Cross-System Work Report screen shown in Figure 18-24 was displayed.

![Figure 18-24 Cross-System Work Report screen](image-url)
Because we again produce two reports with one job, we changed the DDname to something meaningful, selected Processing option 1, provided a report title, and typed line action S next to the Performance selection criteria option.

As we did for the Wait Analysis report, we created two sets of Performance selection criteria, each going to a separate DDname. The first set of selection criteria was for the time range before the application changes were made. The second set was for the time range after the application changes were made. When we completed entering our selection criteria, we pressed F3 repeatedly until we reached the Report Set screen. Then we entered the RUN line action on the Cross-System Work report to submit the job.

When our job was complete, we had the two reports shown in Figure 18-25 and Figure 18-26.

**Figure 18-25  Cross-System Work baseline report**

In Figure 18-25, you see the Cross-System Work report before any application changes were made. Each line is printed from a single CMF performance class record. Records that are part of the same network unit of work are printed sequentially in groups separated by blank lines. All the transactions that are part of our VSAMFSHP object are reported. For transaction BR1, you see that it is always paired with one CSMI mirror task. We took a single unit of work and described the output.
Transaction BR1 was attached by terminal input. It is a user transaction being entered at termid P001. The initial program is BROWSE1. The task’s principal facility is a terminal by name of P001. The task ran on a CICS region with APPLID of SCSCPAA1. It had a task ID of 91968, and its response time was .0066 seconds. While it was running, the application issued EXEC CICS commands to access files. The files were not in the same CICS region, so CICS generated a function shipping request for file control. The function shipping requests were sent over from CICS region SCSCPAA1 to a CICS File Owning Region (FOR) with APPLID SCSCPTA1 to be executed. In this FOR, the mirror task transaction CSMI ran on session X1 as task number 76608, and the response time of these function shipping requests was .0047 seconds.

Cross-System Work report after application changes

Now let's look at the Cross-System Work report in Figure 18-26 after the application changes were made. You see the same kind of pairing of a single BR1 transaction and a single CSMI...
However, for many units of work, you see multiple CSMI tasks. In addition, when there are multiple CSMI tasks associated with a BR1 transaction unit of work, the response times are longer. The interpretation of the data from the BR1 entry and the first CSMI entry shown in area 2 of the report in Figure 18-26 is the same as in Figure 18-25 on page 404 so we only discuss the second CSMI here.

The second CSMI task in area 2 shows that while the application was running, it issued EXEC CICS commands, which caused Interval Control function shipping requests to be sent from region SCSCPTA1 to SCSCPJA7. The Interval Control requests ran the CSMI mirror transaction in SCSCPJA7 on session <AY1 as task number 379.

In the Cross-System Work report from after the application changes, we've seen that, if a unit of work has only a line for the BR1 transaction and a single line for a CSMI transaction, the response times are approximately the same as they were before application changes were made. It is only those units of work where there are multiple CSMI mirror tasks where the function shipping activity takes longer.

Now we want to investigate the function shipping activity in more detail for these tasks.

18.7 Producing the File Usage detailed list report

This section uses the File Usage detailed list report to investigate the type of file access changes that were introduced to our application.

We take a closer look at the function shipped file control requests issued during the BR1 transaction's unit of work. The Cross-System Work reports we produced show us that the file access is performed for our BR1 transaction by the CSMI mirror task in the FOR. The File Usage detailed list report can provide a more granular analysis of file control requests, telling us both which files are accessed as well as what types of access is occurring.

We started by collecting the task numbers for several CSMI tasks appearing in our Cross-System Work report from Figure 18-26 on page 405. We wanted a representative sample from the BR1 units of work that have one CSMI task associated with them, as well as several with multiple CSMI tasks associated with them. We created an Object containing task numbers for CSMI tasks that we want to look at. We did this, because producing the File Usage detailed list report without any selection criteria would produce a much larger report than we wanted to work with. We named our Object List CSMIFSHP, and filled it with the transaction numbers shown in Table 18-2.

<table>
<thead>
<tr>
<th>Single CSMI in BR1 unit of work</th>
<th>Multiple CSMI in BR1 unit of work</th>
</tr>
</thead>
<tbody>
<tr>
<td>82125</td>
<td>81677</td>
</tr>
<tr>
<td>82144</td>
<td>83365</td>
</tr>
<tr>
<td>82910</td>
<td>84105</td>
</tr>
<tr>
<td>84257</td>
<td>84482</td>
</tr>
<tr>
<td>87848</td>
<td>84739</td>
</tr>
<tr>
<td>89126</td>
<td>86766</td>
</tr>
<tr>
<td>90837</td>
<td>90062</td>
</tr>
<tr>
<td>91296</td>
<td>90824</td>
</tr>
<tr>
<td>92138</td>
<td>92783</td>
</tr>
<tr>
<td>92594</td>
<td>94057</td>
</tr>
</tbody>
</table>

We returned to the Report Set screen and typed line action S next to Transaction Resource Usage List report. The screen shown in Figure 18-27 is displayed.
Chapter 18. Using CICS Performance Analyzer reports for problem determination

We selected only the File Usage detailed report, provided a meaningful title, typed line action $ next to Performance selection criteria, and pressed Enter. The screen in Figure 18-28 is displayed.

We left the date and time fields in the Report Interval section blank, and entered an INC selection criteria for task numbers that are in the CSMIFSHP object list that we created. We pressed F3 repeatedly to return to the Report Set screen. Then we entered line action RUN to submit the job. When the job completed, we received the report shown in Figure 18-29.
Report area 1 shows the details for task number 82910, a CSMI for a BR1 task that had only one CSMI task in the unit of work. Area 2 shows the details for task number 83365, a CSMI.
for a BR1 task that had multiple CSMI tasks in the unit of work. Comparing the rest of the transactions in the report, these two seem to be representative of the pattern of file use.

By comparing the file access patterns, you quickly see that both tasks are browsing file VENDORX. Our tasks with the higher response time are making 151 browse calls to file control 4, while our quickly responding tasks only make four 3.

This file is in an FOR and is remote from where the application program making the request is running. The new changes to program BROWSE1 are causing many more browse requests on the VENDORX file, and these requests are all being function shipped from SCSCPA1A1 to SCSCPTA1. Function shipping browse requests causes two flows in the network for each record requested. For large numbers of calls, the overhead can be unacceptably high.

18.8 The second CSMI mirror task

Next we investigate the reason for the second CSMI mirror task associated with these longer running BR1 transactions. We already know that the additional function shipping for browse requests on the VENDORX file are responsible for the longer IRIOWTT suspend. But we wanted to see more details about the CSMI task that is involved in interval control.

To do so, we ran a list report using one of the supplied sample report forms. From the CICS PA Primary Option Menu, we selected option 3 to go to the Report Forms screen. From there, we could either enter the SAMPLES command and press Enter or use the SAMPLES action bar choice and choose option 1 from the list. The Sample Report Forms screen in Figure 18-30 is displayed.

Figure 18-30 Sample Report Forms

We pressed F8 to page down through the list until we reached the ICLST sample form. We typed line action S next to it and pressed Enter. Then we pressed F3 to save our selection.
We returned to the Report Set screen, typed line action $ next to the List report, and pressed Enter. We already produced the List report before, so a line entry for a List report is displayed on the screen. We typed line action $ and pressed Enter. The screen in Figure 18-31 is displayed.

Figure 18-31 Performance List report screen

We selected APPLID SCSCPJA7, since we are only concerned about mirror tasks in that region.

We placed the cursor in the Form field and pressed F4 to select from the list of available report forms.

We use the S line action and specify transaction CSMI and a small time interval in order to reduce the amount of output produced.

Figure 18-32 Interval Control List Report

We find task number 379, which also appeared in our Cross-System Work report in Figure 18-26 on page 405. We see that this task issued one interval control start command. We went back to the application program and found that an EXEC CICS START TRAN command was added as part of the application changes. Our BR1 transaction ran in
SCSCPA1, but the transaction that was being started ran in SCSCPJA7, so CICS had to run a mirror task in SCSCPJA7 to start the new application transaction.

All of the CSMI transactions which ran in SCSCPJA7 during our time interval were for interval control start commands. The additional mirror task was not responsible for adding significant suspend time so we did not investigate it further.

At this point, we have enough information to discuss with the applications programming group responsible for program BROWSE1. We can discuss what the program changes were supposed to accomplish and if other more efficient methods exist to accomplish the same task. If there are no programmatic alternatives, there are still some options that the CICS Systems Programmer can take to improve the situation.

### 18.9 Options to improve transaction BR1 performance

This section discusses two different options the CICS Systems Programmer has to improve the response time for transaction BR1.

#### 18.9.1 Define the VENDORX file as a CICS Maintained Data Table

If the application cannot perform its task without browsing the VENDORX file, then the CICS Systems Programmer has a couple of options open to them. The first option is to eliminate the function shipping activity for read requests on the VENDORX file.

Our past experience with the VENDORX file indicates that a high percentage of the file requests are for read-only access. This type of access pattern is one of the things that makes it a good candidate for a CICS Maintained Data Table (CMDT). A major benefit of making VENDORX a CMDT is that function shipping is avoided for most read and browse requests. Without function shipping, we eliminate the IRIOWTT MRO link wait time for file control requests from our response time for transaction BR1.

To accomplish this change and measure the impact, we changed the VENDORX File definition to make it a CMDT in region SCSCPTA1. We ran the same TPNS workload and produced a Performance Summary report. Figure 18-33 shows that report.

![Figure 18-33 Performance Summary with VENDORX as a CMDT](image-url)
In the report, the average response time for transaction BR1 was reduced to .0097 from the high of .0446 seconds in Figure 18-13 on page 397. Also the average Dispatch Wait time, which increased by 1450%, is now comparable to what it was before any application changes were made. And the average IR Wait Time is now less than its original value from Figure 18-12 on page 396.

### 18.9.2 Define the VENDORX file locally

In the CMDT scenario, we let CICS load the entire VENDORX file into a data table. For large application files, this may not be a viable option. Another option that you can choose to help improve transaction BR1 response time is to move the VENDORX file out of the FOR and back into SCSCPA1. This eliminates all function shipping for VENDORX file requests from applications in this AOR.

To accomplish this change and measure the impact, we changed the VENDORX File definition to make it a local file in SCSCPAA1. We then ran the same TPNS workload and produced a Performance Summary report. Figure 18-34 shows that report.

Now the average response time for transaction BR1 is reduced to .0090 from the high of .0446 seconds in Figure 18-13 on page 397. Also the average Dispatch Wait time and the average IR Wait Time are back to their levels before any application changes were implemented.

An added benefit of moving the file to the AOR is that the average response time for transaction SX4, which updates the VENDORX file, is reduced by 63% from what we saw in Figure 18-12 on page 396.

### 18.10 Future problem determination efforts

We created several reports, all of which are useful in diagnosing future performance problems with CICS. Instead of running them individually in separate jobs, you can run all reports in a single pass of SMF data. This is the benefit of the Report Set.

We specified the report interval in the selection criteria for each report. To use these reports as a set in the future, we don’t want to have to update each report with new time ranges. CICS PA allows you to specify the time range independent of each individual report.
From the CICS PA Primary Option Menu, we selected option 2 and then typed line action S next to the report set we created to return to the Report Set screen. We typed line action S next to each report we created and updated the performance selection criteria to remove only the report interval, leaving any transaction or object list specification alone. We returned to the Report Set screen and typed line action S next to the global performance selection criteria as highlighted in bold in Figure 18-35.

![Figure 18-35 Specifying Performance selection criteria for a Report Set](image)

By specifying a performance selection criteria date and time range in this way, we can apply it to all reports in the report set rather than specifying it individually on each report. The next time we need to run this series of reports, we can provide the appropriate date and time criteria here. Then we can use the RUN command from the command line instead of the RUN line action on each individual report.

### 18.11 Conclusion

Using a series of CICS Performance Analyzer reports, we examined the individual components of the response time for an application transaction, and quantitatively measured the impact made by an application program change. We drilled down to find the cause of a performance problem, made system configuration changes, and measured the impact of those changes.
Historical Database

CICS Performance Analyzer (PA) V1.3 introduces the Historical Database (HDB), which enables you to collect and manage historical performance data for your CICS systems. It is a repository of CICS-related performance data by which you can maintain a history of CICS transactions. Reporting can be done on a list or a summary base. For more details, see Part 4 “Using the Historical Database (HDB)” in the CICS Performance Analyzer for z/OS User's Guide, SC34-6307.

This chapter explains how to use the Historical Database. We concentrate on the summary Historical Database. We work through a workload scenario and show the different functions and reports that are available in the HDB.
19.1 Introduction to the HDB

Figure 19-1 provides a visual overview of the HDB. The register is a VSAM KSDS that can contain information about several different HDBs. An HDB is a description of data that is loaded in containers. The data in the containers is copied from CICS system management facility (SMF) records. The most important characteristics that are kept in an HDB tell which type of data is collected, summary or list. They also indicate which SMF fields are copied to the container records, the filtering criteria, the location of the data sets that represent the containers, and how long the containers must exist before deletion.

![Figure 19-1 Overview of the HDB](image)

19.2 System setup and scenario overview

Figure 19-2 shows the system setup of the Teleprocessing Network Simulator (TPNS) environment and the CICS regions that we used in this scenario.

![Figure 19-2 System setup for DB2 workload](image)
On MVS image SC47, we used TPNS to simulate a 3270 workload on the shown CICS regions in MVS image SC66.

PTA1 and PTA2 are the SYSIDs of two TOR regions. PJA6 and PJA7 are the AORs. PTA1 does static routing of all incoming transactions to PJA7. PTA2 does static routing to PJA6. The two AORs are connected to the same DB2 system.

In a first run, three DB2 transactions are executed:

- DB2N executes program PROGDB2N, which executes 15 fetches from the DB2 provided sample table DSN8710.EMP. Between each fetch, there is a non-threadsafe EXEC CICS command. PROGDB2N is defined with CONCURRENCY(QUASIRENT).

- DB2R executes program PROGDB2R, which executes 15 fetches from the DB2 provided sample table DSN8710.EMP. There are no non-threadsafe EXEC CICS commands in this program so that it is defined with CONCURRENCY(THREADSAFE).

- DB2U executes program PROGDB2U that is an update application. It sends a BMS map to the 3270 screen. It gives the option to cancel the application or to perform a retrieve and update on the PHONENO field of table DSN8710.EMP. PROGDB2U is defined with CONCURRENCY(THREADSAFE).

In a later phase of the scenario, a fourth transaction is added. DB2V executes program PROGDB2V, which is a second update application. It contains only one SQL statement that performs an update of the PHONENO field of table DSN8710.EMP. PROGDB2V is defined with CONCURRENCY(QUASIRENT).

### 19.3 CICS changes

Authorized program analysis report (APAR) PQ76703 adds new monitoring and statistics functions to CICS TS 2.2. One of the new options specifies whether you want additional monitoring performance class data to be collected for the resource managers used by your transactions. This way, the time spent in the External Resource Managers (ERM) is added in different new Resource Manager Interface (RMI) fields in the SMF record. This option is activated via a new RMI parameter on the TYPE=INITIAL macro of the monitoring control table (MCT).

Before running this scenario, we assembled the MCT that is shown in Example 19-1. It shows the specification of the RMI=YES parameter and the suffix that we must specify for the MCT parameter in the SIT or SYSIN overriding of CICS.

#### Example 19-1  MCT with RMI=YES

<table>
<thead>
<tr>
<th>RMI</th>
<th>DFHMCT TYPE=INITIAL,</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>APPLNAME=YES,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILE=10,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMI=YES,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUFFIX=RM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this scenario, we include the DB2 RMI field in the reports that we print with CICS PA.
19.4 Using the HDB

This chapter goes through the different options that are introduced by the Historical Database Manager. It starts by showing the different steps we had to go through to load and report from a Summary HDB. Then it discusses additional maintenance and housekeeping functions. Finally it uses a List HDB to demonstrate the export function that allows you to load the data from an HDB into a DB2 table.

19.4.1 Summary HDB

The usage of a summary HDB is similar to using performance summary reports. The records that are written to the HDB summarize transaction activity over a specified time interval. This reduces the amount of data written to the HDB. By measuring a set of transactions that issue DB2 SQL requests, we show you how to load, report, and interpret summary data.

From the CICS Performance Analyzer Primary Option Menu, we entered option 5 to select Historical Database as shown in Figure 19-3.

![Primary Option Menu: Selecting Historical Database](image)

The Historical Database Menu (Figure 19-4) that opens shows the facilities of the Historical Database Manager.

![HDB primary menu](image)

You must specify the name of the HDB register before you enter any option from the HDB menu. The HDB register is the inventory of all information associated with the CICS PA Historical Database Manager. The HDB register contains HDB definitions, data set definitions for HDB containers and the HDB templates. You can define as many registers as required.
However, only one register can be used at a time. We recommend that you define only one register. This allows users to share performance data.

As shown in Figure 19-5, we selected option 1 to create an HDB template and specified the name of the HDB register we wanted to use. Since this is the first action, the HDB register must still be created. CICS PA prompts for the required information to create the VSAM KSDS for the HDB register. As suggested in the CICS Performance Analyzer for z/OS User’s Guide, SC34-6307, we specified the allocation unit of cylinders with a primary and secondary quantity of 1. We decided that the HDB register had to be allocated on volume TOTSTR.

--- Historical Database ---

Command ===>

Enter "/* to select option
Edit IDCAMS command
Browse errors only

HDB Register Name . . 'CICSRS4.CICSPA.HDB.DB2TREND'

Cluster Level Information:

Space Units ..... 1
1. Cylinders Primary Quantity . 1
2. Tracks Secondary Quantity . 1
3. Records
4. Kilobytes
5. Megabytes

Volume . . . . . . . . TOTSTR
Data Class . . . .
Management Class . .
Storage Class . .

--- Figure 19-5 HDB allocation information ---

CICS PA automatically allocates the VSAM data set for the HDB and shows the IDCAMS output, as shown in Figure 19-6.

--- Figure 19-6 IDCAMS output from the DEFINE CLUSTER of the HDB register ---

We pressed F3 to go to the empty HDB Templates screen (Figure 19-7). HDB templates allow you to define the type and format of the data in the HDB. This is similar to the way Report
Forms define the type and format of the data in a report. Using the NEW command, we entered the name of the template that we were going to define, which was DB2TRNDS.

```
File Options Help
-----------------------------------------------
HDB Templates
Command ==> NEW DB2TRNDS  Scroll ==> CSR
Select to edit Template. Enter NEW command to define a new Template.
/ Name     Type      Description     Changed     ID
******************************************* End of list *******************************************
```

Figure 19-7  HDB Templates screen to create the DB2TRNDS template

CICS PA presents a pop-up screen (Figure 19-8), where we selected the type of template that we wanted to use. We selected option 2 for a summary template because we wanted to look for a trend in the DB2 transactions.

```
--------------------- Historical Database ---------------------
File Systems Options Help
------------------------------------------------------------
New HDB Template
Command ==>
Specify the name of the new Template and its options.
Name . . . . DB2TRNDS
APPLID . . . + Version (VRM) . . +
MVS Image . . Field Categories
Type 2  1. List
      2. Summary
```

Figure 19-8  Selecting a summary template

The next screen asks for the contents of the template. On this screen, we edited the template format and content to meet our specific reporting and analysis requirements. For this summary template, we chose to have the fields as shown in Figure 19-9. Notice also that we typed line action S next to the Performance selection criteria option. Selection criteria can be specified to apply filtering to the data being collected by the HDB template.
Presuming that CICS is running all day, we entered a five minute time interval from a peak period in the Performance Select Statement screen (Figure 19-10). We did not specify a date, because we intended to look at an extract of the SMF records that is taken every day at the same time. We chose only to have reports about transactions with names that start with DB2.
This ends the creation of the summary template. We pressed F3 until we reached the Historical Database Menu, where we entered option 2 to define the HDB itself. As shown in Figure 19-11, we entered the name and description, the name of the template to be used, the data retention period, and the information for the storage space allocation and attributes for the container data set that will hold the data. Since this is a summary HDB, we did not have to allocate big data sets.

**Figure 19-10  DB2TRNDS selection criteria**

**Figure 19-11  DB2TRNDS HDB definition**
We pressed F3 to create the definition. Back on the Historical Database Menu, we selected option 3 to load the data into a container. Figure 19-12 shows the Load HDBs screen with our first HDB, which we selected.

![File Systems Options Help](image)

Select to load an HDB.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2TRNDS</td>
<td>SUMMARY</td>
<td>DB2TREND HDB</td>
<td>2003/10/02 13:23</td>
<td>CICSRS4</td>
</tr>
</tbody>
</table>

Figure 19-12  Selecting DB2TRNDS HDB on the Load HDBs screen

Figure 19-13 shows Load Summary screen which is displayed next. We did not complete any parameters on this screen. We pressed Enter twice to see the generated JCL of the load job.

![File Systems Options Help](image)

Specify HDB load options then press Enter to continue submit.

System Selection: MM/DD/YYYY HH:MM:SS.TH
APPLID . . + From
Image . . SC66 + To
Group . . +

Enter "/" to select option
/ Edit JCL before submit

Figure 19-13  Load SUMMARY for DB2TRNDS HDB

The JCL in Example 19-2 shows that we are using multiple SMF input data sets. In our scenario, the SMF data sets filled up and switched several times. The original SMF data sets were archived to the data sets that you see in the different SMFINxx data sets. The archived data sets are in a generation data group (GDG) that contains 30 data sets. SMF data that must be kept for a longer time requires a manual copy. After loading the HDB, we no longer needed these data sets because all the reporting came from the data stored in the HDB.

Example 19-2  Generated JCL for loading the DB2TRNDS HDB

```
// JOB
// * CICSPA V1R3 HDB LOAD JCL
// CICSPA EXEC PGM=CPAMAIN
// STEPLIB DD DSN=CPA13.SCPALINK,
// DISP=SHR
// CPAHDBRG DD DSN=CICSRS4.CICSPA.HDB.DB2TREND,
// DISP=SHR
// SYSPRINT DD SYSOUT=*```
/* SMF Input Files */
//SMFIN001 DD DSN=SMFDATA.CICSRECS.G0409V00,
//            DISP=SHR  
//SMFIN002 DD DSN=SMFDATA.CICSRECS.G0410V00,
//            DISP=SHR  
//SMFIN003 DD DSN=SMFDATA.CICSRECS.G0411V00,
//            DISP=SHR  
//SMFIN004 DD DSN=SMFDATA.CICSRECS.G0412V00,
//            DISP=SHR  
//SMFIN005 DD DSN=SMFDATA.CICSRECS.G0413V00,
//            DISP=SHR  
//SMFIN006 DD DSN=SMFDATA.CICSRECS.G0414V00,
//            DISP=SHR  
//SMFIN007 DD DSN=SMFDATA.CICSRECS.G0415V00,
//            DISP=SHR  
//SMFIN008 DD DSN=SMFDATA.CICSRECS.G0416V00,
//            DISP=SHR  
/* Command Input */
//SYSIN DD *
* Report Set=DB2TRNDS 
* Description=CICSPA HDB request 
* Reports for Image=SC66 
* Description=System added by take-up 
CICSPA IN(SMFIN001,
SMFIN002,
SMFIN003,
SMFIN004,
SMFIN005,
SMFIN006,
SMFIN007,
SMFIN008),
NOAPPLID,
LINECNT(60),
FORMAT(':','/'),
HDB(OUTPUT(HDBL0001),LOAD(DB2TRNDS))
*/

Notice that the time selection that we entered in the template Performance select statement in Figure 19-10 on page 422 is kept in the HDB. It is not visible in the CICS PA commands. If you need to change the time limits, you go again through the CICS PA screens to update the template selection criteria before you submit the job.

We entered the SUB command on the JCL edit screen to submit the load job. Figure 19-14 shows the recap report of the load job.
Only 62 records were written to the allocated container. Because this is a summary container, all average values are calculated before loading the HDB. For a list HDB, the fields that are specified in the list template are stored in one record per transaction.

We produced a report from the data that is now available in the HDB. Using the summary template, we selected several fields that we thought to be necessary for our trend follow up. However, as in other reports, the number of fields printed is determined by the maximum page width of 132 characters. Therefore, we left the HDB Manager. From the Primary Option Menu, we selected the report forms option 3 to create a summary report form. If we did not provide a report form, the order of the printed fields would be as specified in the template but up to 132 characters. The report form that we created shown in Figure 19-15 is named DB2SUM.

![Figure 19-15 Summary Report Form for use with the HDBTRNDS HDB](image)

This being done, we returned to the HDB Manager. On the Historical Database Menu, we selected option 4, which brought us to the HDB Reporting screen (Figure 19-16).
We selected the DB2TRNDS Historical Database. Then the Run SUMMAR HDB Report screen (Figure 19-17) is displayed. We entered the name of the summary report form that we created and left all other fields on their default value.

Example 19-3 shows the resulting JCL that is generated. At the bottom of the JCL, you can see a DD statement for the container data set that is used for printing the requested report. If there were more containers, they would all be listed in this generated JCL. As the comments in the JCL explain, this DD statement is there only for reference and is not required.

Example 19-3  Generated JCL for printing the DB2TRNDS HDB

```
//  JOB
//  */ CICSPA V1R3 HDB Report JCL
//CICSPA   EXEC PGM=CPAMAIN
//STEPLIB  DD DSN=CPA13.SCPALINK,
//          DISP=SHR
//CPAHDBRG DD DSN=CICSPA.HDB.DB2TREND,
//          DISP=SHR
//SYSPRINT DD SYSOUT=*
//  */ Command Input
```
After submitting this JCL, we received our first report shown in Figure 19-18 and Figure 19-19 on page 429.

You can ignore the first lines of this report. They show a start time of 10:54. These are transactions that started before but ended within the time interval that we specified. We do not print them again in the following reports.

Within the time interval, you see, for each APPLID, a list of the selected transactions. After each group of transactions, a total for this group is inserted and at the end of each time interval an additional grand total is printed. Note that these totals can be left out by selecting the Exclude Totals option on the Run SUMMARY HDB Reporting screen.

This is a rather long report because this example shows a one minute default time interval. Bigger time intervals result in less data written to the container and therefore smaller reports. An optimal reporting interval must be found between the amount of data loaded and the level of detail of the report. We decided that for the next results, we would report on the five minutes interval that we collect data for our measurements.
### Historical Database Summary

#### Start Time

- **2003/10/02 10:54**
- **2003/10/02 10:55**
- **2003/10/02 10:56**
- **2003/10/02 10:57**

#### DB2U
- **SCSCPJA7**: Task 2
- **SCSCPJA6**: Task 1

#### DB2R
- **SCSCPJA7**: Task 1
- **SCSCPJA6**: Task 1

#### DB2N
- **SCSCPJA7**: Task 1
- **SCSCPJA6**: Task 1

---

**Figure 19-18** Summary report output of the DB2TRNDS HDB (Part 1 of 2)
Loading the HDB with summary records can be a daily process so that a trend in transaction behavior is easy to follow in the report. A graph produced with these daily results would show severe fluctuations when they would appear.

For the next load of data to the HDB, we could have gone to the Systems Definition Menu of CICS PA and assigned a set of new SMF input data sets to system SC66. Then we again could have used the load option on the Historical Database Menu to generate a new load JCL and submit the job.

However, when we submitted the load job the first time, we decided to save the JCL in a separate JCL library. CICS PA uses a temporary library for generating the JCL. You find the name of this data set on the EDIT line of the ISPF panel (Figure 19-20).
In ISPF split screen mode, we edited a new member from a JCL library and copied the content of the temporary data set containing the load JCL as shown in Example 19-2 on page 423 into our private library. As a second step of this job, in the same way, after the copy of the load job, we copied the JCL of the report job that is shown in Example 19-3 on page 426. The resulting job is shown in Example 19-4.

Because all information is contained in the HDB, there is no need to change the HDB control statement of the load step. We only had to change the SMF input data set names and eventually adapt the list of SMFINxxx DD cards in the CICS PA IN command.

As mentioned earlier, we decided to change the time interval for the report to 5 minutes. To do so, we manually changed the INTERVAL parameter of the report step.

Finally, we commented out the last line of the report step. The DD statement is there as reference for the container being used to print the report but is not required. We commented it out to avoid a JCL error in case the container would no longer exist.

The next day, when new SMF data sets became available, we submitted this job to produce a second report.

Example 19-4   Two step load and report job to be submitted from a private library

```
//CICSR4L JOB (ACCOUNT), 'NORBERT', MSGLEVEL=(1,1), NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC66
//*/CICSPA V1R3 HDB LOAD JCL
//CICSPA EXEC PGM=CPAMAIN
//STEPLIB DD DSN=CPA13.SCPALINK,
// DISP=SHR
//CPAHDDBRG DD DSN=CICSR4S.CICSPA.HDB.DB2TRND,
// DISP=SHR
//SYSPRINT DD SYSOUT=*
/* SMF Input Files
//SMFIN001 DD DSN=SMFDATA.CICSRECS.G0449V00,
// DISP=SHR
//SMFIN002 DD DSN=SMFDATA.CICSRECS.G0450V00,
// DISP=SHR
//SMFIN003 DD DSN=SMFDATA.CICSRECS.G0451V00,
// DISP=SHR
//SMFIN004 DD DSN=SMFDATA.CICSRECS.G0452V00,
// DISP=SHR
```
The HDB LOAD Recap Report in Figure 19-21 shows the new container data set name and the updated time range of the data.
LOAD requested for HDB: DB2TRNDS  Register DSN: CICSRS4.CICSPA.HDB.DB2TRNDS

The following Container(s) were created and loaded:
Container DSN: CICSRS4.DB2TRNDS.D03276.T150925.HDB  No of Records: 83
Start Timestamp: 2003-10-03-10.54.00  End Timestamp: 2003-10-03-10.59.00

LOAD process complete.

**Figure 19-21** HDB LOAD Recap Report from the second data load

**Figure 19-22** shows the new report that is based on a reporting interval of five minutes.

---

**Figure 19-22** HDB report containing the second day
This report shows one total per transaction per CICS region. We see that a new transaction, DB2V, was added to the running set of applications. Looking at the summary records for DB2N, DB2R and DB2U, we see that the second set of data shows an increase in the average response and suspend time.

We used Lotus 1-2-3 to visualize these reports by means of some charts. The bars in the charts represent the results from our first and second run ordered by transaction ID. Knowing that TOR PTA1 is routing all transactions to AOR PJA7, the graph in Figure 19-23 shows that the increase in PTA1 came from an increase in PJA7.

Figure 19-23  TOR PTA1 - AOR PJA7 response time comparison

Figure 19-24 shows that in PJA7 the dispatch time also increased for the three transactions. The CPU time only increased for DB2N. For the two other transactions, there is no difference. This comparison did not lead to a conclusion.

Figure 19-24  PJA7 Dispatch time and CPU time comparison

The new DB2 RMI time shown in Figure 19-25 shows that the increased elapsed time came from the longer time that we spent in the DB2 attachment.
In the first instance, we looked at the program that was added and executed by running the new DB2V transaction. This program contains an SQL update command followed by an EXEC CICS SUSPEND command. This program is defined with CONCURRENCY(QUASIRENT). The plan of the program is bound with RELEASE(COMMIT). This means that the SUSPEND command, although being a threadsafe command, runs under the QR TCB. The SQL update results in a DB2 update lock.

The result of the SUSPEND command is that control is passed to the dispatcher so that another task of higher or equal priority can process. When this is the case, DB2V remains suspended on the QR TCB while holding the DB2 update lock. This causes other transactions to remain suspended longer within DB2 until DB2V gets control again. After the SUSPEND, the program ends and the update lock is released. We removed the EXEC CICS SUSPEND command from the program code. A new set of data was collected from the available SMF records and was added to the HDB. The resulting report is shown in Figure 19-26 and Figure 19-27.
| Interval | Start | APPLID | Tran | Tasks | Response | Dispatch | User | CPU | Susp | LB CPU | DispWait | DB2 | ChngMode | MaxOTDly | RMI | DB2 | Avg | Time | Avg | Avg | Avg | Avg | Avg | Avg |
|----------|-------|--------|------|-------|----------|----------|------|-----|------|--------|----------|-----|----------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2003/10/03 10:55 | SCSCPJA6 DB2N | 5799 | 0.0224 | 0.0081 | 0.0027 | 0.0142 | 0.0117 | 0.0119 | 17 | 36 | 0.0033 | 0.0067 |
| 2003/10/03 10:55 | SCSCPJA6 DB2R | 7094 | 0.0125 | 0.0075 | 0.0022 | 0.0050 | 0.0016 | 0.0029 | 17 | 4 | 0.0033 | 0.0064 |
| 2003/10/03 10:55 | SCSCPJA6 DB2U | 32458 | 0.0072 | 0.0041 | 0.0009 | 0.0031 | 0.0004 | 0.0013 | 1 | 2 | 0.0002 | 0.0033 |
| 2003/10/03 10:55 | SCSCPJA6 DB2V | 2311 | 0.0126 | 0.0082 | 0.0012 | 0.0044 | 0.0008 | 0.0022 | 1 | 4 | 0.0005 | 0.0075 |
| 2003/10/03 10:55 | SCSCPJA6 DB2Z | 47662 | 0.0101 | 0.0053 | 0.0013 | 0.0048 | 0.0008 | 0.0029 | 5 | 7 | 0.0002 | 0.0044 |
| 2003/10/03 10:55 | SCSCPJA7 DB2N | 5752 | 0.0211 | 0.0085 | 0.0026 | 0.0136 | 0.0017 | 0.0116 | 17 | 36 | 0.0022 | 0.0071 |
| 2003/10/03 10:55 | SCSCPJA7 DB2R | 6940 | 0.0123 | 0.0079 | 0.0027 | 0.0044 | 0.0015 | 0.0028 | 17 | 4 | 0.0002 | 0.0067 |
| 2003/10/03 10:55 | SCSCPJA7 DB2U | 32768 | 0.0069 | 0.0042 | 0.0009 | 0.0028 | 0.0004 | 0.0012 | 1 | 2 | 0.0001 | 0.0034 |
| 2003/10/03 10:55 | SCSCPJA7 DB2V | 2334 | 0.0125 | 0.0085 | 0.0013 | 0.0039 | 0.0008 | 0.0021 | 1 | 4 | 0.0002 | 0.0080 |
| 2003/10/03 10:55 | SCSCPJA7 DB2Z | 47794 | 0.0098 | 0.0054 | 0.0013 | 0.0044 | 0.0007 | 0.0022 | 5 | 6 | 0.0002 | 0.0045 |
| 2003/10/03 10:55 | SCSCPJA1 DB2N | 5749 | 0.0342 | 0.0140 | 0.0004 | 0.0272 | 0.0000 | 0.0020 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA1 DB2R | 6939 | 0.0247 | 0.0105 | 0.0004 | 0.0232 | 0.0000 | 0.0028 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA1 DB2U | 32765 | 0.0191 | 0.0083 | 0.0013 | 0.0173 | 0.0000 | 0.0021 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA1 DB2V | 2334 | 0.0242 | 0.0102 | 0.0003 | 0.0229 | 0.0000 | 0.0019 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA1 DB2Z | 47787 | 0.0220 | 0.0087 | 0.0003 | 0.0203 | 0.0000 | 0.0022 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 DB2N | 5798 | 0.0339 | 0.0140 | 0.0003 | 0.0325 | 0.0000 | 0.0020 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 DB2R | 7092 | 0.0248 | 0.0105 | 0.0004 | 0.0233 | 0.0000 | 0.0026 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 DB2U | 32454 | 0.0199 | 0.0083 | 0.0003 | 0.0181 | 0.0000 | 0.0022 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 DB2V | 2313 | 0.0257 | 0.0140 | 0.0003 | 0.0243 | 0.0000 | 0.0021 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 DB2Z | 47657 | 0.0226 | 0.0107 | 0.0003 | 0.0210 | 0.0000 | 0.0022 | 0 | 0 | 0.0000 | 0.0000 |
| 2003/10/03 10:55 | SCSCPJA2 | 190900 | 0.0161 | 0.0035 | 0.0008 | 0.0126 | 0.0004 | 0.0025 | 2 | 3 | 0.0001 | 0.0022 |

Figure 19-27  HDB report of third run (Part 1 of 2)

The report shows that the transaction suspend times are again down to their normal values. The graphs in Figure 19-28 through Figure 19-30 on page 436 give a more visual representation that we returned to the situation from before the DB2V transaction was added and that the DB2V, after correction, has much less influence on the other transactions.
Figure 19-28  PTA1-PJA7 response times after DB2V correction

Figure 19-29  PJA7 response and CPU times after DB2V correction

Figure 19-30  PJA7 - DB2 RMI time after DB2V correction
19.4.2 List HDB

The usage of a list HDB is similar to using performance list reports. The records that are written to the HDB represent single transaction records. The selected fields of each CMF performance record are copied as one record in the HDB. This way large amounts of data can be copied to the HDB. Thus the list HDB is considered to have a short life span and should be used for detailed analysis of recent transaction events.

The creation of templates, the definition of the HDB, and loading and reporting a list HDBs are similar to the usage of the summary HDB. We show the usage of a list HDB in 19.4.5, “Export HDB data sets to DB2” on page 441.

19.4.3 HDB maintenance

HDB maintenance allows you to change the options or to flag containers from the HDB as deleted. We entered option 5 to enter the HDB maintenance option. Then, we were presented the list of the HDBs. Figure 19-31 shows that we selected the DB2TRNDS HDB.

<table>
<thead>
<tr>
<th>File</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HDB Maintenance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command ==&gt;</td>
<td>Scroll ==&gt; CSR</td>
</tr>
</tbody>
</table>

HDB maintenance HDB definitions list

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2TRNDL LIST</td>
<td>LIST</td>
<td>DB2trndl HDB</td>
<td>2003/09/27 15:07</td>
<td>CICSRS4</td>
</tr>
<tr>
<td>s DB2TRNDS SUMMARY</td>
<td>SUMMARY</td>
<td>DB2trnds HDB</td>
<td>2003/09/29 11:24</td>
<td>CICRS54</td>
</tr>
</tbody>
</table>

Figure 19-31 HDB maintenance HDB definitions list

We selected an HDB and then had two view options. First is the Maintain options view where you can alter the options of the HDB (Figure 19-32).
Second is the Maintain data sets view. You can request to have this view by changing the Specify View value or by pressing F11, which allows you to switch between the two views. Figure 19-33 shows the list of allocated containers in the HDB. It shows that the first container is expired.
We selected the third container to see additional details about the container itself and about the data that is available in it. Figure 19-34 shows this detail.

![Figure 19-34 HDB Maintenance: Container details](image)

We pressed F3 to return to the previous screen where we entered a D in front of the container name to flag it as deleted. Figure 19-35 shows the deleted status of the selected container. You can undo this status by typing line action U next to the container name. Undo is available until you run housekeeping.

![Figure 19-35 HDB Maintenance: An expired and deleted container](image)

19.4.4 HDB housekeeping

HDB housekeeping allows you to generate JCL to physically delete the containers that are flagged as expired or deleted. It also allows you to generate JCL for doing an IDCAMS VERIFY on a broken HDB register KSDS. This can be required if you receive message IEC161I after a failed HDB dialog or batch request.

When we selected housekeeping, the screen shown in Figure 19-36 is displayed.
Figure 19-36  HDB Housekeeping options

Figure 19-35 shows an expired and a deleted container. To physically delete them, we selected option 1 on the HDB Housekeeping screen to run the HDB housekeeping job. Example 19-5 shows the two step JCL that was submitted.

Example 19-5  HDB housekeeping JCL

```plaintext
// JOB
// /* CICSPA VIR3 HDB HOUSEKEEPING JCL
// HKEEP EXEC PGM=CPAMAIN
// STEPLIB DD DSN=CPA13.SCPALINK,
//          DISP=SHR
// CPAHDBRG DD DSN=CICSRS4.CICSPA.HDB.DB2TRND,
//           DISP=SHR
// SYSPRINT DD SYSOUT=*  
// SYSSIN DD *  
//   CICSPA HDB(HKEEP)/*
// CPAHKDEL DD DSN=&CPAHKDEL,DISP=(NEW,PASS),
//         SPACE=(CYL,(1,1))/*
// DELETE EXEC PGM=IDCAMS,COND=(0,NE,HKEEP)
// SYSPRINT DD SYSOUT=*  
// SYSSIN DD DSN=&CPAHKDEL,DISP=(OLD,DELETE) 
```

Figure 19-37 shows the output of the two steps of the HDB housekeeping job. It also shows that the DB2TRNDL HDB had two expired containers. This job should be submitted periodically to physically delete the expired containers.
We selected option 2 on the HDB Housekeeping menu for the Repair HDB Register using the VERIFY command. No job was submitted but the VERIFY command was executed dynamically. The result is shown on the Historical Database Menu (Figure 19-38).

![Figure 19-37 HDB housekeeping output](image)

19.4.5 Export HDB data sets to DB2

To perform more extended analysis on SMF data, you may try loading SMF data into DB2 tables. The Export function of the Historical Database Manager allows you in an easy way to...
generate the required jobs to create the DDL to define a DB2 table and a job to load the data into this table.

We show you how we used a list HDB to load a DB2 table. Since it is very to create the required jobs to define a table and load a DB2 table, it is easier to work with small tables that contain only the data that you require. However, to understand our storage requirements, we decided to create a table with all available columns.

First we created a list template with name DB2ALL. On the Historical Database Menu, we selected option 1 to enter the templates option. We entered the NEW command and then entered the name of the template that we wanted to create. A pop-up screen appeared where we selected to have a list template. Figure 19-39 shows that we selected all fields to be included in the template by moving the End of HDB marker to the last line of all selectable fields.

![Figure 19-39 HDB template with all fields selected](image)

Then, on the main menu, we selected option 2 to define the HDB. Figure 19-40 shows the HDB definition panel with the options we specified. This HDB is only used briefly so we specified a retention period of one day. We specified a specific volume serial because we were sure there was enough space available on that volume. For primary space allocation, we specified 30 cylinders. The secondary allocation was set to 5.
Next we loaded the data into the HDB. We selected the load option 3 and selected the just created DB2ALL HDB. Figure 19-41 shows that we selected to have only the SMF records for APPLID SCSCPJA7. We pressed Enter to preview the JCL which we then submitted.

Figure 19-42 shows part of the job log and the recap report of the load job. The abend B37 indicates that the storage space allocation for the data container was too small. This did not cause a problem for us. When CICS PA detects an abend x37, it closes the file and continues loading data in a new allocated container. This job had an abend B37 resulting in two containers being allocated for holding all of the data. The fact of having two containers...
instead of one is transparent to the report function but not to the export function that can export only one container at a time.

Figure 19-42  Job log and Load recap Report showing an abend B37

We returned to the main HDB menu and selected option 5 to create the jobs for loading the container data into a DB2 table. We selected our DB2ALL HDB and were presented the screen as shown in Figure 19-43. We selected the first container and pressed Enter.

Figure 19-43  Available containers for DB2ALL Export HDB function

Figure 19-44 shows the screen that we received with the selected container name. This screen is used twice. First we selected option 1 to create the DDL to define the DB2 table.

Next for the required information about the DB2 system that we were going to use, we entered CPA13 as the name of the DB2 database that has to be created to contain our DB2 table. We did not enter a VCAT value because we decided to use the default storage group, SYSDEFLT. We also entered the primary and secondary allocation units for the DB2 tables.

Finally, we chose to have both time and count values for the timer fields. A CMF performance class clock field consists of a timer value and a count value. The timer value represents the total time value as it was accumulated during one or more measurement periods. The count value gives the number of these different periods. We did not select the Include Sums of Squares option since this does not apply to a list HDB.
Example 19-6 shows the job that was generated. We recognized our parameters in the CREATE DATABASE, TABLESPACE, and TABLE commands. For a typical field, such as CPU time, we saw that two columns are included: CPU_TIME and CPU_COUNT. At the end of the job, we noticed also that an index is built for the table.

Example 19-6  DB2 DDL for creating a table to load HDB data into

```sql
// JOB /*JOBPARM SYSAFF=SC66 /* CICSPROF V1R3 HDB - DDL TO DEFINE DB2 TABLE /*RUNTIAD EXEC PGM=IKJEFT01,DYNAMNBR=20 /*STEP1 LIB DD DISP=SHR,DSN=DB7Q7.SDSNLOAD /* DD DISP=SHR,DSN=DB7Q7.SDSNEXIT /*SYSTSPT DD SYSCOUT="*" /*SYSTSIN DD * DSN SYSTEM(D7Q2) RUN PROGRAM(DSNTIAD) - LIB('DB7Q7.RUNLIB.LOAD') PLAN(DSNTIA71) /**/ /*SYSPRINT DD SYSCOUT="*" /*SYSDUMP DD SYSCOUT="*" /*SYSDUMP DD * CREATE DATABASE CPA13;
COMMIT;
```
CREATE TABLESPACE DB2ALL
  IN CPA13
  LOCKSIZE ANY
  BUFFERPOOL BPO
  CLOSE NO
  SEGSIZE 32
  USING STOGROUP SYSDEFLT
  PRIQTY 5000
  SECQTY 500
  ERASE NO;

CREATE TABLE CPA13.DB2ALL (
  START TIMESTAMP,
  TMVSID CHAR(4),
  APPLID CHAR(8),
  TRAN CHAR(4),
  USERID CHAR(8),
  PROGRAM CHAR(8),
  TASKNO INTEGER,
  RESPONSE_TIME FLOAT,
  DISPATCH_TIME FLOAT,
  DISPATCH_COUNT INTEGER,
  CPU_TIME FLOAT,
  CPU_COUNT INTEGER,
  SUSPEND_TIME FLOAT,
  SUSPEND_COUNT INTEGER,
  DISPWAIT_TIME FLOAT,
  DISPWAIT_COUNT INTEGER,
  FCWAIT_TIME FLOAT,
  FCWAIT_COUNT INTEGER,
  ...
  WBTOTAL INTEGER,
  WBWRITE INTEGER
) IN CPA13.DB2ALL;

CREATE TYPE 2 UNIQUE INDEX CPA13.DB2ALL_IX
  ON CPA13.DB2ALL
  (START,
   MVSID,
   APPLID,
   TRAN,
   USERID,
   PROGRAM
  )
  USING STOGROUP SYSDEFLT
  PRIQTY 10
  SECQTY 10
  ERASE NO
  CLUSTER
  BUFFERPOOL BPO
  CLOSE NO;
We submitted this job and returned to the Export screen (Figure 19-44). We then chose option 2 to generate the job for loading the data into the newly created table. Example 19-7 shows the generated JCL and the first lines of the LOAD commands.

**Example 19-7  Generated job for loading the HDB data into the generated DB2 table**

```plaintext
//CICSRS4 JOB (ACCOUNT),'NORBERT',MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC66
//* CICSPA V1R3 HDB - LOAD DATA INTO DB2 TABLE
//DSNUPROC EXEC PGM=DSNUTILB,REGION=0M,
// PARM='D7Q2'
//STEPLIB DD DISP=SHR,DSN=DB7Q7.SDSNLOAD
// DD DISP=SHR,DSN=DB7Q7.SDSNEXIT
//SYSPRINT DD SYSOUT=* 
//UOPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSREC DD DSN=CICSRS4.DB2ALL.DO32B1.T155543.HDB, 
// DD DISP=SHR 
//SYSUT1 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 
//SORTOUT DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 
//SYSIN DD * 
LOAD DATA RESUME YES 
INTO TABLE CPA13.DB2ALL ( 
START                POSITION(1)     TIMESTAMP EXTERNAL(26), 
MVSID                POSITION(27)    CHAR(4), 
APPLID               POSITION(31)    CHAR(8), 
TRAN                 POSITION(39)    CHAR(4), 
. . .
```

We modified the SYSREC DD statement to concatenate the second container that was created when the data was loaded to the HDB.

**Attention:** Be careful when concatenating the container data sets as described here. Ensure that the template has not changed between times.

We submitted the job with the concatenated containers but received a DB2 abend S04E. Figure 19-45 shows the error messages. Reason code 00D70014 indicates the failure of an attempt to extend the tablespace.

```plaintext
DSNU398I -D7Q2 DSNUWRBF - UNEXPECTED PROCESSING ERROR, REASON=X'00E40318' ON TABLE - CPA13.DB2ALL
DSNT500I  DSNUGBAC - RESOURCE UNAVAILABLE 
REASON 00D70014 
TYPE 00000220 
NAME DB7QU.DSNDBC.CPA13.DB2ALL.I0001.A001 
DSNU017I    DSNUGBAC - UTILITY DATA BASE SERVICES MEMORY EXECUTION ABENDED, REASON=X'00E40318'
```

**Figure 19-45  DB2 error messages from the DB2 load job**

At this time, we were not able to perform any action on the DB2 resources because the DB2 load utility was still in progress. We entered the DB2I interface in TSO as shown in Figure 19-46 to determine the exact status of the load utility and to take an appropriate action to remove this status. We selected option 7 to enter the DB2 commands option.
We entered the DISPLAY UTILITY command to check the status of our load job.

```
Figure 19-46  DB2I Primary Option Menu: selecting DB2 Commands

Command ==> 7

Select one of the following DB2 functions and press ENTER.

1  SPUFI         (Process SQL statements)
2  DCLGEN        (Generate SQL and source language declarations)
3  PROGRAM PREPARATION (Prepare a DB2 application program to run)
4  PRECOMPILE     (Invoke DB2 precompiler)
5  BIND/REBIND/FREE (BIND, REBIND, or FREE plans or packages)
6  RUN           (RUN an SQL program)
7  DB2 COMMANDS  (Issue DB2 commands)
8  UTILITIES     (Invoke DB2 utilities)
D  DB2I DEFAULTS (Set global parameters)
X  EXIT         (Leave DB2I)
```

```
Figure 19-47  DB2 Commands: DISPLAY UTILITY

Example 19-8 shows the reply of the DISPLAY command.

```
Example 19-8  Output of the DISPLAY UTILITY command

<table>
<thead>
<tr>
<th>DSNU100I</th>
<th>-D7Q2 DSNUDIS - USERID = CICSRS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBER</td>
<td>D7Q2</td>
</tr>
<tr>
<td>UTILID</td>
<td>CICSRS4.CICSRS4L</td>
</tr>
<tr>
<td>PROCESSING</td>
<td>UTILITY STATEMENT 1</td>
</tr>
<tr>
<td>UTILITY</td>
<td>LOAD</td>
</tr>
<tr>
<td>PHASE</td>
<td>RELOAD  COUNT = 0</td>
</tr>
<tr>
<td>NUMBER</td>
<td>OBJECTS IN LIST = 1</td>
</tr>
<tr>
<td>LAST</td>
<td>OBJECT STARTED = 1</td>
</tr>
<tr>
<td>STATUS</td>
<td>STOPPED</td>
</tr>
<tr>
<td>DSN9022I</td>
<td>-D7Q2 DSNUGCCC '-DIS UTIL' NORMAL COMPLETION</td>
</tr>
</tbody>
</table>
```

The utility output shows that our load utility is in stopped status. We copied the UTILID value and used it in the second DB2 command that is shown in Figure 19-48. With the cursor on the second command line, we pressed Enter to run the TERMINATE command.
Chapter 19. Historical Database

Figure 19-48  DB2 commands: TERMINATE the load utility

Figure 19-49 shows the output of the TERMINATE command.

The utility was terminated, so we could drop the DB2 objects. Still in the DB2I application, we entered SPUFI and executed the following commands:

```
DROP TABLE CPA13.DB2ALL;
DROP TABLESPACE CPA13.DB2ALL;
DROP DATABASE CPA13;
```

Then we returned to the Export HDB screen (Figure 19-44 on page 445) and changed the primary allocation value for the tablespace to 50000 and the secondary value to 5000. We re-ran the DDL job to create the DB2 table and the load job for the DB2 table. This time, both jobs ran without problems. Figure 19-50 shows the last DB2 messages of the load job output.

The output shows that the table was loaded with 59956 records, which is the sum of the number of records in the two data containers.

```
DSNU320I -D7Q2 DSNURWI - RESUME(YES) WAS SPECIFIED FOR EMPTY TABLESPACE
DSNU304I -D7Q2 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=59956 FOR TABLE CPA13.DB2ALL
DSNU302I -D7Q2 DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:01:46
DSNU042I -D7Q2 DSNUGSOR - SORT PHASE STATISTICS - NUMBER OF RECORDS=59956
DSNU349I -D7Q2 DSNURBXA - BUILD PHASE STATISTICS - NUMBER OF KEYS=59956 FOR INDEX CPA13.DB2ALL_IX
DSNU258I -D7Q2 DSNURBXD - BUILD PHASE STATISTICS - NUMBER OF INDEXES=1
DSNU259I -D7Q2 DSNURBXD - BUILD PHASE COMPLETE, ELAPSED TIME=00:01:00
DSNU010I -D7Q2 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

Figure 19-50  Load job output messages

We again entered SPUFI and executed a SELECT on SYSIBM.SYSTABLES to look for the characteristics of our table. The output in Figure 19-51 shows that our table has 334 columns.
Figure 19-51   DB2ALL table information

To look at the contents of our table, we ran a SELECT * SQL statement. Figure 19-52 shows this SQL statement as well as the first screen of output.

```sql
SELECT * FROM CPA13.DB2ALL
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>CREATOR</th>
<th>TYPE</th>
<th>DBNAME</th>
<th>TSNNAME</th>
<th>DBID</th>
<th>OBID</th>
<th>COLCOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2ALL</td>
<td>CPA13</td>
<td>T</td>
<td>CPA13</td>
<td>DB2ALL</td>
<td>269</td>
<td>3</td>
<td>334</td>
</tr>
</tbody>
</table>

Figure 19-52   SQL SELECT command output

To have a decimal representation instead of a floating-point representation of the time fields, as an example, we selected specific fields and asked for a decimal representation. For such fields, we also re-specified the column name. Example 19-9 demonstrates how we did this.

**Example 19-9   Table content with decimal number representation**

```sql
SELECT TRAN,
 TASKNO,
 DEC(RESPONSE_TIME,8,4) AS RESPONSE_TIME,
 DEC DISPATCH_TIME,8,4 AS DISPATCH_TIME,
 DISPATCH_COUNT,
 DEC CPU_TIME,8,4 AS CPU_TIME
FROM CPA13.DB2ALL
WHERE TRAN LIKE ('DB2_')
ORDER BY RESPONSE_TIME DESC
```

<table>
<thead>
<tr>
<th>TRAN</th>
<th>TASKNO</th>
<th>RESPONSE_TIME</th>
<th>DISPATCH_TIME</th>
<th>DISPATCH_COUNT</th>
<th>CPU_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2N</td>
<td>39974</td>
<td>.6276</td>
<td>.5533</td>
<td>38</td>
<td>.0034</td>
</tr>
<tr>
<td>DB2R</td>
<td>39972</td>
<td>.5887</td>
<td>.5533</td>
<td>7</td>
<td>.0027</td>
</tr>
<tr>
<td>DB2U</td>
<td>39970</td>
<td>.5885</td>
<td>.5411</td>
<td>7</td>
<td>.0014</td>
</tr>
<tr>
<td>DB2U</td>
<td>39969</td>
<td>.5877</td>
<td>.5507</td>
<td>6</td>
<td>.0019</td>
</tr>
<tr>
<td>DB2U</td>
<td>39968</td>
<td>.5873</td>
<td>.5499</td>
<td>6</td>
<td>.0017</td>
</tr>
<tr>
<td>DB2U</td>
<td>39966</td>
<td>.5583</td>
<td>.5509</td>
<td>5</td>
<td>.0018</td>
</tr>
<tr>
<td>DB2V</td>
<td>39965</td>
<td>.5581</td>
<td>.4989</td>
<td>6</td>
<td>.0017</td>
</tr>
<tr>
<td>DB2N</td>
<td>39962</td>
<td>.5521</td>
<td>.4644</td>
<td>37</td>
<td>.0035</td>
</tr>
<tr>
<td>DB2N</td>
<td>71697</td>
<td>.5508</td>
<td>.0440</td>
<td>37</td>
<td>.0035</td>
</tr>
<tr>
<td>DB2N</td>
<td>71698</td>
<td>.5481</td>
<td>.4010</td>
<td>37</td>
<td>.0036</td>
</tr>
</tbody>
</table>
19.5 Conclusion

This chapter showed how you can use the Historical Database. The creation, loading, and reporting from a summary HDB was demonstrated by going through an error scenario. The way the data is accumulated in an HDB allows for an easy way to detect eventual changes in the performance of your transactions or complete CICS system.

We used a list HDB to show the export function in more detail. The export function provides a flexible way to load selected CICS performance data into a DB2 table.

We also showed the easy-to-use maintenance and housekeeping functions of the Historical Database Manager.
# Abbreviations and acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACB</td>
<td>access control block</td>
</tr>
<tr>
<td>ACID</td>
<td>atomicity, consistency, isolation, durability</td>
</tr>
<tr>
<td>AID</td>
<td>automatic initiate descriptor</td>
</tr>
<tr>
<td>AOR</td>
<td>application owning region</td>
</tr>
<tr>
<td>APAR</td>
<td>authorized program analysis report</td>
</tr>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>APPC</td>
<td>advanced program-to-program communication</td>
</tr>
<tr>
<td>ARM</td>
<td>automatic restart manager</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>ATI</td>
<td>automatic transaction initiation</td>
</tr>
<tr>
<td>BMS</td>
<td>basic mapping support</td>
</tr>
<tr>
<td>BTS</td>
<td>business transaction services</td>
</tr>
<tr>
<td>CAS</td>
<td>coordinating address space</td>
</tr>
<tr>
<td>CF</td>
<td>coupling facility</td>
</tr>
<tr>
<td>CICS</td>
<td>Customer Information Control System</td>
</tr>
<tr>
<td>CICS PA</td>
<td>CICS Performance Analyzer for OS/390</td>
</tr>
<tr>
<td>CICS TS</td>
<td>CICS Transaction server</td>
</tr>
<tr>
<td>CMAS</td>
<td>CICSPlex System Manager address space</td>
</tr>
<tr>
<td>CMF</td>
<td>CICS Monitoring Facility</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CSI</td>
<td>common system area</td>
</tr>
<tr>
<td>CSD</td>
<td>CICS system definition</td>
</tr>
<tr>
<td>CTG</td>
<td>CICS Transaction Gateway</td>
</tr>
<tr>
<td>CUA</td>
<td>Common User Access</td>
</tr>
<tr>
<td>CWA</td>
<td>common work area</td>
</tr>
<tr>
<td>CWS</td>
<td>CICS Web Support</td>
</tr>
<tr>
<td>DASD</td>
<td>direct access storage device</td>
</tr>
<tr>
<td>DBCTL</td>
<td>database control</td>
</tr>
<tr>
<td>DCT</td>
<td>destination control table</td>
</tr>
<tr>
<td>DPL</td>
<td>distributed program link</td>
</tr>
<tr>
<td>DSA</td>
<td>dynamic storage area</td>
</tr>
<tr>
<td>EBCIDIC</td>
<td>Extended Binary Coded Decimal Interchange Code</td>
</tr>
<tr>
<td>ECB</td>
<td>event control block</td>
</tr>
<tr>
<td>ECI</td>
<td>external call interface</td>
</tr>
<tr>
<td>EDSA</td>
<td>extended dynamic storage area</td>
</tr>
<tr>
<td>EIB</td>
<td>EXEC interface block</td>
</tr>
<tr>
<td>EIP</td>
<td>EXEC interface program</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise JavaBean</td>
</tr>
<tr>
<td>ELPA</td>
<td>extended link pack area</td>
</tr>
<tr>
<td>EMP</td>
<td>event monitoring point</td>
</tr>
<tr>
<td>EPI</td>
<td>external presentation interface</td>
</tr>
<tr>
<td>ESDS</td>
<td>entry sequenced data set</td>
</tr>
<tr>
<td>ESM</td>
<td>external security manager</td>
</tr>
<tr>
<td>EXCI</td>
<td>external CICS interface</td>
</tr>
<tr>
<td>FEPI</td>
<td>front-end programming interface</td>
</tr>
<tr>
<td>FOR</td>
<td>file owning region</td>
</tr>
<tr>
<td>GLUE</td>
<td>global user exit</td>
</tr>
<tr>
<td>GMT</td>
<td>Greenwich Mean Time</td>
</tr>
<tr>
<td>GTF</td>
<td>Generalized Trace Facility</td>
</tr>
<tr>
<td>GUI</td>
<td>graphical user interface</td>
</tr>
<tr>
<td>HDB</td>
<td>Historical Database</td>
</tr>
<tr>
<td>HFS</td>
<td>hierarchical file system</td>
</tr>
<tr>
<td>HLL</td>
<td>high level language</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>hypertext transfer protocol</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>ICF</td>
<td>Integrated Catalog Facility</td>
</tr>
<tr>
<td>IDE</td>
<td>integrated development environment</td>
</tr>
<tr>
<td>IIO</td>
<td>Internet Inter-ORB Protocol</td>
</tr>
<tr>
<td>IMS</td>
<td>Information Management System</td>
</tr>
<tr>
<td>IPCS</td>
<td>Interactive Program Control System</td>
</tr>
<tr>
<td>IPL</td>
<td>initial program load</td>
</tr>
<tr>
<td>IRC</td>
<td>interregion communication</td>
</tr>
<tr>
<td>ISC</td>
<td>intersystem communication</td>
</tr>
<tr>
<td>ISPF</td>
<td>Interactive System Productivity Facility</td>
</tr>
<tr>
<td>ISV</td>
<td>Independent Software Vendor</td>
</tr>
<tr>
<td>IT</td>
<td>information technology</td>
</tr>
<tr>
<td>ITSO</td>
<td>International Technical Support Organization</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>JCL</td>
<td>job control language</td>
</tr>
<tr>
<td>JCT</td>
<td>journal control table</td>
</tr>
<tr>
<td>JDK</td>
<td>Java Development Kit</td>
</tr>
</tbody>
</table>
JNI  Java native interface
JVM  Java Virtual Machine
KSDS  key sequenced data set
LE  Language Environment
LPA  link pack area
LPAR  logical partition
LSR  local shared resources
LU  logical unit
LUW  logical unit of work
MAS  managed address space
MCT  monitoring control table
MRO  multiregion operation
MVS  multiple virtual storage
OLTP  online transaction processing
ORB  Object Request Broker
OTE  Open Transaction Environment
PDS  partitioned data set
PDSE  partitioned data set extended
PLT  program list table
PLTP  program list table post initialization
PLTSD  program list table shutdown
PTF  program temporary fix
RACF®  Resource Access Control Facility
RDO  resource definition online
RLS  record level sharing
RMF  Resource Management Facility
RRDS  relative record data set
SIT  system initialization table
SMF  system management facility
SMS  storage management subsystem
SNA  Systems Network Architecture
SOS  short on storage
SQL  Structured Query Language
SSL  Secure Sockets Layer
SVC  supervisor call
TCB  task control block
TCP/IP  Transmission Control Protocol/Internet Protocol
TCT  terminal control table
TCTTE  terminal control table terminal entry
TCTUA  terminal control table user area
TD  transient data
TIOA  terminal input output area
TOR  terminal owning region
TPNS  Teleprocessing Network Simulator
TRUE  task related user exit
TS  temporary storage
TWA  transaction work area
UOW  unit of work
URL  Uniform Resource Locator
URM  user replaceable module
VSAM  Virtual Storage Access Method
VTAM  Virtual Telecommunications Access Method
WLM  workload management
WUI  Web User Interface
XML  Extensible Markup Language
XPI  exit programming interface
Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 456. Note that some of the documents referenced here may be available in softcopy only.

- Accounting and Chargeback with Tivoli Decision Support for OS/390, SG24-6044
- CICS Transaction Gateway V5: The WebSphere Connector for CICS, SG24-6133
- Enterprise JavaBeans for z/OS and OS/390 CICS Transaction Server V2.2, SG24-6284
- DB2 for z/OS and OS/390 Tools and Performance Management, SG24-6508
- DB2 Table Editor Tool Version 4.2, SG24-6833

Other publications

These publications are also relevant as further information sources:

- OS/390 V2R10.0 MVS Workload Management Services, GC28-1773
- OS/390 V2R10.0 MVS System Management Facilities (SMF), GC28-1783
- CICS Transaction Server for z/OS Migration Guide, GC34-5984
- z/OS V1R4.0 MVS System Management Facilities (SMF), SA22-7630
- DB2 UDB for OS/390 and z/OS Administration Guide, SC26-9931
- z/OS Resource Management Facility Report Analysis, SC33-7991
- CICS Transaction Server for z/OS CICS System Definition Guide, SC34-5988
- CICS Transaction Server for z/OS CICS Customization Guide, SC34-5989
- CICS Transaction Server for z/OS CICS Resource Definition Guide, SC34-5990
- CICS Transaction Server for z/OS CICS Operations and Utilities Guide, SC34-5991
- CICS Transaction Server for z/OS CICS Supplied Transactions, SC34-5992
- CICS Transaction Server for z/OS CICS Application Programming Guide, SC34-5993
- CICS Transaction Server for z/OS CICS Application Programming Reference, SC34-5994
- CICS Transaction Server for z/OS CICS System Programming Reference, SC34-5995
Online resources

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

- CICS TS V2.2 online library

- CICS Performance Analyzer online Library

- Fundi Software

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](http://ibm.com/redbooks)

Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)
Index

Numerics
3270 Bridge
  interface 257
  two task model 80

A
  abend B37 443
  accounting
    CLASS 1 15
    class 1, class 3 15
    CLASS 2 15
  ACCOUNTREC parameter 179
  ACCOUNTREC(TASK) 15
  ACCOUNTREC(UOW 15
  activity keypoint interval 236
  advanced program-to-program communication (APPC) 5
  AKPFREQ 226, 234
  APAR
    PQ33097 263
    PQ34321 268
    PQ38838 321
    PQ63141 26, 139, 311
    PQ63143 26, 139, 311
    PQ63527 26, 311
    PQ76695 139
    PQ76701 139
    PQ76703 207
    PQ79058 26
  API 7
  APPC connections 243
  Application Naming 311
  APPLD 90
  APPLID (MULTIPLE 101
  APPLID/MVS ID 100
  auto-journals 224
  auxiliary storage 216

B
  BTE execution phase 379
  BTS report 35, 61, 81
  Business Transaction Services report 82
  BWO facility 9

C
  CALL command 136
  capacity planning 16
  CBLPSHPOP 320
  CEDX 251
  CEECCICS 320
  CEECOPT 320
  CEEDOPT 320
  CEEROPT 320
  CEEUOPT 321
  CEEWCOPT 320
  CEMT
    INQUIRE JVM 303
    INQUIRE MONITOR 10
    INQUIRE TCPIPSERVICE 253
    SET DISPATCH 282
    SET DSN QUIESCE ALL 157
    SET MONITOR 10
  CEMT SET MONITOR 164
  CETR transaction 324
  CICS
    CICS-DB2 Attachment Facility xiii, 470
    CICS-VSAM interface xiii, 470
    data and utilities 17
    DB2 task-related user exit (TRUE) 173
    GTF data 22
    monitoring 10
    statistics 18
    CICS and DB2 connectivity 173
    command threads 173
    entry threads 173
    pool threads 173
  CICS Business Event Publisher for MQSeries 3–4, 8
  CICS Business Transaction Services (BTS) 81
  CICS DB2 Attachment Facility
    functions 172
    overview 172
    reporting 179
  CICS Interdependency Analyzer 3, 8
  CICS Monitoring Facility
    data fields 48
    data transfer to SMF 10
    data types 10
    DFH$MCTD 14
    dictionary record 34
    exception class data 10–11, 25
    exception class records 51
    output processing 14
    performance class data 10–11, 25
    performance data 25
    performance records 44
    transaction resource class data 10, 12, 25–26
    user fields 34
  CICS Online Transmission Time Optimizer 3
  CICS Performance Analyzer
    Active status 41
    and Transaction Resource Monitoring 165
    benefits 7
    Class 3 list report sorted by transaction 213
    commands 64
      for HDB processing 64
      for reports and extracts 64
    comparison with CICS PM 8
    creating Report Sets data sets 35
CICS Performance Analyzer

data used 10
default report format 65
defining a new system 32
dialog 25
dictionary 34
Edit Object List 56
Edit Report Form 46
Exception reports 61, 84
Exclude Totals option 427
exporting HDB data sets to DB2 441
extracts 63
EXWAIT field 344
Function keys 26
Global Options 38
HDB housekeeping 439
HDB maintenance 437
importing extracted data
  into Lotus 1-2-3 110
  into Microsoft Access 118
  into Microsoft Excel 113
invocation 29
JCL edit 429
JCL generation 42
List Export 69
maintaining Object Lists 56
MCT specification 34
Object Lists 56, 146
Performance extracts 99
Performance Graph reports 97
Performance List Extended report 67
Performance List report 65
Performance reports 65
Performance select statement 292
Performance Summary report 70
Performance Totals report 72
preparing the SMF data 28
Primary Option Menu 29, 143, 389
Processing extracts with Microsoft Excel 112
Profile Options Menu 29
Recap report 77, 179
Record Selection 271
report format 73
Report Set 60
report tailoring 68
reporting 34
reporting CICS DB2 Attachment Facility 179
Reports list 40
requesting reports and extracts 35
running reports 40
sample forms 273
selection criteria 347
specifying report options 39
submitting the Report Set 40
Subsystem reports 62
Summary Export 72
System Definitions Menu 30
system grouping 33
System Logger report 96
tailoring Report Forms 44
take-up 30, 389
job 271
Take-Up from SMF 143, 159
Take-up function 208
Transaction Group report 261
Transaction Resource Usage report 62
Wait Analysis report 75
Web report 261
working with different CICS releases 358

CICS Performance Analyzer commands
  CROSS 64
  DB2 64
  EXPORT 64
  FIND command 47
  GRAPH(RESPONSE) 64
  GRAPH(TRANRATE) 64
  LIST 64
  LISTEXC 64
  LISTX 64
  LOGGER 64
  MQ 64
  NEW command 32
  RECSEL 64
  RESUSAGE(FILESUM) 64
  RESUSAGE(TRANLIST) 64
  RESUSAGE(TSSUM) 64
  RUN command 54
  SAMPLES command 45
  SUMEXC 64
  SUMMARY 64
  TOTAL 64
  TRANGROUP 64
  VERIFY command 441
  WAITANAL 64
  WORKLOAD 64

CICS Performance Analyzer commands for HDB processing
  HDB(HKEEP) 65
  HDB(LOAD(hdbname)) 64
  HDB(REPORT(hdbname)) 65

CICS Performance Analyzer DB2 Reports content 179
  Buffer Manager statistics 179
  Class 1 Thread elapsed and CPU times 179
  Class 2 In-DB2 elapsed and CPU times 179
  Class 3 Suspend times 179
  DB2 Thread Identification 179
  Locking statistics 179
  SQL DML statistics 179

CICS Performance Analyzer fields
  APPLPROG 312
  APPLTRAN 312
  Count with DASD Write 227
  Count without DASD Write 227
  DB2CONWT 178, 192, 197
  DB2RDYQW 178
  DB2REQCT 178
  DB2WAIT 179
  DISPWAIT 178
  EXWAIT 344
  MAXJTDLY 279
  ORIGIN 261
RMI MQ time field 216
RMISUSP 178
RMITIME 178
TDWAIT 48

CICS Performance Analyzer functions
FIELDS/BY 312
Maintain SMF files 249
Take-up from SMF 179

CICS Performance Analyzer Report Forms
TDLST 45
TDSUM 45

CICS Performance Analyzer reports
BTS report 61, 81
Business Transaction Services 82
Cross-System Work Extended 377
Cross-System Work extract 63, 80
Cross-System Work report 61, 77
UOWs selection 375
DB2 List 179
DB2 report 62, 89
DB2 Summary 179
Exception List report 84, 353
Exception reports 61, 84
Exception Summary report 85, 353

exclusion list
CFOR - RLS quiesce 363
CFQS - RLS quiesce 363
CKTI - MQSeries default listener task initiator 363
COHT - CICSPlex SM completed task history recorder 363
COIO - CICSPlex SM CMAS communication task 363
COIE - CICSPlex SM heartbeat task 363
CONL - CICSPlex SM MAS initialization and control transaction 363
COMN - CICSPlex SM monitoring task 363
CSHQ - Scheduler services 363
CSKL - TCP/IP default Sockets listener 363
CSNC - IRC connection manager 363
CSNE - VTAM node abnormal 363
CSOL - TCP/IP listener 363
CSSY - CICS internal task 363
CSTP - CICS internal task 363
CSZI - FEPI 363

Export extract 63
File Usage Summary report 62, 86
List report temporary storage usage of MQ transactions 216
Logstream Summary report 232
Performance Graph 97
Performance List Extended report 65, 68
Performance List report 61, 65
Performance selection criteria 261
Performance Summary report 61, 65
Performance Totals report 61, 65
Recap report 179
Record Selection extract 63
Report Set screen 260
sample reports
Report Form CPULST 359

SSTORLST 263
USTORLST 263
System Logger - Logstream Summary report 96
System Logger report 94
Temporary Storage Usage Summary report 62, 87
Transaction Group option 260
Transaction Group report 61
Transaction Rate Graph report 63, 97
Transaction Resource Usage List report 62, 88
Transaction Response Time Graph report 63, 98–99
Wait Analysis report 61
section 1 402
section 2 402
WebSphere MQ report 62, 91
Workload Activity report 61, 82, 378
Workload Activity Summary report 378

CICS Performance Monitor
benefits 7
collection with CICS PA 8
history definitional 6
monitoring component 6
reporting component 6
server component 6
threshold definitional component 6
view sets
CICS dispatcher 6
CICS region 6
collection storage (DSA and subpool usage) 6
DB2 connection 6
enqueue pools 6
global and task related user exits 7
JVM pool 7
loader 6
recovery manager, including UOW analysis 6
temporary storage 6
transaction manager 6
transient data 6
workstation client 6

CICS resources
connections 7
files, data tables, and local shared resource (LSR) pools 7
journals 7
log streams 7
programs 7
temporary storage queues 7
terminals 7
transaction classes 7
transactions 7
transient data queues 7

CICS SMF records 416
CICS statistics data
DFH0STAT 19
end-of-day 18
interval 18
processing 19
requested 18
requested reset 18
unsolicited 18
CICS system definition after take-up update 160
CICS Performance Analyzer

CICS tools
- CICS Business Event Publisher for MQSeries 3, 8
- CICS Interdependency Analyzer 3–4, 8
- CICS Online Transmission Time Optimizer 3–4, 8
- CICS Performance Analyzer 4
- CICS Performance Monitor 4
- CICS VSAM Recovery 3, 9
- IBM Session Manager for z/OS 3, 9
- CICS Transaction Gateway xiii, 241–242
- CICS Transaction Server xiii, 470
- V1.3 xiii, 470
- V2.2 xiii, 470
- CICS TS V2.2 195
- CICS VSAM interface 140
- CICS VSAM Recovery 3, 5, 9
- CICS Web Support (CWS) 80
- CICS/ESA V4.1 242
- CICS/VSE 2.3 242
- CICSPFOR 34
- CICSPlex SM 130
  - address space (CMAS) 130
  - Web User Interface (WUI) 6
- CICSPlex System Manager (CICSPlex SM) 6
- CICSVR 9
- CIEP transaction 253
- CIRP transaction 300, 376
- Class 1 91
- Class 3 91
- Class 3 List report
  - detailed look at MQ activity 211
  - sorted by transaction 213
  - -classpath parameter 246
- CLER transaction 321
- client daemon 242
- CLIENTCLASSPATH variable 246
- CLOCK operand 323
- CMF 6
  - data fields 48
  - dictionary 358
  - dictionary record 34, 358
  - exception class data 61
  - exception class records 51
  - fields 35, 53, 100
  - input data 77
  - performance class clock field 444
  - performance class data 61–63
    - summarized by transaction and program 184
  - performance class record 35, 51, 65, 67, 70, 73, 78, 90, 207
  - performance data 105
  - performance fields 359
  - performance record 44, 359
  - making a subset 356
  - suspend components 77
  - transaction resource class data 62
  - transaction resource class record 51
  - user fields 34–35
- CMF monitor data
  - exception class data 10

- COBOL 245, 386
- command threads 173
- COMAREA 245
- Common User Access (CUA) 9
- CONCURRENCY parameter 194
- CONCURRENCY(THREADSAFE) parameter 176
- connections 7
- CORBA 268

- correlating performance data
  - between CICS and DB2 90, 186, 197
  - between CICS PA and DB2 PM 38
  - between multiple CICS systems 61
  - by CICS BTS process ID 61
  - by network UOW ID 12
  - by transaction group 80
  - by transaction sequence number 12
  - Cross-System Work Extended report 377
  - Cross-System Work report 77
  - in a Cross-System Work report 61
  - Workload Activity report 378

- coupling facility
  - data table 85
  - log structure 224
- CPA0311E 359
- CPMI transaction 255
- CPU 105
- CPU and Dispatch statistics 73
- CPUSUM report form 368
- CREA transaction 287
- CROSS command 64
- Cross-System Work Extended report 377
- Cross-System Work extract 35, 63, 99
- Cross-System Work report 35, 61, 77, 376, 403
  - default format 78
  - tailoring 79
- CSMI transaction 249–250
- CSOL transaction 258, 363
- CSQ4CVJ1 206
- CSQ4CVK1 206
- CTG is V5.0.1 242
- CTG_RRMNAME environment variable 255
- CUA (Common User Access) 9
- CWBA transaction 263
- CWS URL 258
- CWXN storage consumption 263
- CWXN transaction 257, 260, 266

D
- DASD offloading 224
- DASD shift 224, 227, 233
- DASD-only 224
- DB2 6, 24
  - abend S04E 447
  - attachment 433
  - CICS-DB2 Attachment Facility xiii, 470
  - command 448
  - database 444
  - messages 449
objects 449
RMI 433
SQL 418
table 449
tablespace 444
update 434
DB2 accounting data 14
DB2 accounting trace 14
DB2 Attachment Facility
API 172
attachment commands 172
DB2 commands 172
DB2 authorization checking 172
DB2 command 64
DB2 command thread 173
DB2 Entry threads 173
DB2 List report 186
DB2 Performance Monitor (DB2 PM) 90
DB2 Performance Monitor for OS/390 5
DB2 Plan name 90
DB2 PM 38, 90
DB2 report 35, 62, 89–90
List 197
Long Summary 185, 193, 197
Recap 179
Short Summary 184, 192, 197
DB2 SQL Performance Analyzer for OS/390 5
DB2 SSID (subsystem ID) 182
DB2 table 451
DB2 time stamps 38
DB2 V7.1 address space 171
DB2CONN
ACCOUNTREC parameter 15
definition 176, 194
DB2ConWt field 192
DB2ENTRY
ACCOUNTREC parameter 15
definition 179, 187, 193–194
PROTECTNUM parameter 173
THREADLIMIT parameter 173
DB2 interface 447
DB2Wait field 177
DBCTL 13, 67
statistics 67
DDL 444
Debug Tool for z/OS and OS/390 5
DELETEQ TS 217
DF$JVM group 269
DF$MCTD 14
DF$MOLS 5, 20
DF$SOT 263
DF$STAT 5, 19
DF$APPL 26
DF$BWTTA 258
DF$CSDUP 18
DF$JUP 17, 21
DF$LOG 225–227, 233
DF$SCU 228, 236
DFMCT 12, 14, 139
DFMCTAS 14
DFMCTDS 14
DFMCTF$ 14
DFMCTTS 14
DFMNDUP 20, 35
DFMNRRDS 164
DFRMI 26
DFSHUNT 226, 233
DFHST parameter 251
DFHSTUP 5, 19
DFHWBTTA 258, 263
DFJSTS1 program 269
DFSMS 140
DFSMSdss 9
DFSMShs 9
dictionary 358
dictionary record 325, 358, 360
Distributed Program Link (DPL) 78
distributed routing 131
DSECT 19
DSNC 172
DSNCb 172
DSNCLI 173, 287
duplexed 224
dynamic transaction backout 224
E
ECI interface 131, 243
ECI over TCP/IP 246
ECI_EXTENDED parameter 255
ECI_LUW_NEW parameter 252
ECI_NON_EXTENDED parameter 255
ECIRequest 132
ECIRequest Java class 245
Edit Report Form 46
EMP (event monitoring point) 12, 312, 321
Enterprise JavaBean 268
definition 268
entity bean 269
in CICS 135
session beans 268
workloads 130
entry threads 173
EPI interface 131, 243
ESI interface 243
event monitoring point (EMP) 12, 312, 321
user-defined in MCT 73
exception class data 11, 61
exception class records 344
Exception List report 61, 84, 353
Exception records 11
Wait for auxiliary temporary storage 11
Wait for auxiliary temporary storage buffer 12
Wait for auxiliary temporary storage string 12
Wait for coupling facility data tables locking (request) slot 12
Wait for coupling facility data tables non-locking (request) slot 12
Wait for file buffer 12
Wait for file string 12
Wait for LSRPOOL string 12
Wait for storage in the CDSA 11
Wait for storage in the ECDSA 11
Wait for storage in the ERDSA 11
Wait for storage in the ESDSA 11
Wait for storage in the EUDSA 11
Wait for storage in the RDSA 11
Wait for storage in the SDSA 11
Wait for storage in the UDSA 11
Exception reports 61, 84
Exception List 61
Exception Summary 61
exception selection criteria 51
Exception Summary report 61, 85, 353
EXCI calls 131
EXCI interface 131, 246
Exclude Totals option 427
EXE execution phase 380
EXEC CICS
  ASSIGN PROGRAM 318
  COLLECT STATISTICS 19
  HANDLE 320
  IGNORE 320
  INQUIRE MONITOR 10
  INQUIRE STATISTICS 19
  LINK 268
  MONITOR 323
  POP HANDLE 320
  PUSH HANDLE 320
  RETURN 320
  SET MONITOR 10, 312
  START TRANSACTION 268
  SUSPEND 434
Export extract 63, 101
External Call Interface (ECI) over TCP/IP 80
External Resource Managers (ERM) 417
extract record format 100, 104
extract record tailoring 103
extracts 63
  Cross-System Work 80, 99
  DB2 example 202
  Export 101
  importing data into Lotus 1-2-3 110
  importing into Microsoft Access 118
  importing into Microsoft Excel 113
  LINK/CALL example 340
  processing with Lotus 1-2-3 108, 332
  processing with Microsoft Access 118
  processing with Microsoft Excel 112
  Record Selection 104
  VSAM example 156
  EXWAIT field 344
  EYU9XLOP 131, 300
F
File Control (F) 78
File Usage detailed list report 406
File Usage Summary report 62, 86, 167
filtering
  see selection criteria
  the report 51
FIND command 53
  forward recovery logs 224
  function shipping 140
  function shipping request 78
G
gateway daemon 242
Generalized Trace Facility (GTF) 22
GET requests 213
GETMAIN 265
getcode 206
global options 37
group of systems 42, 361
GTF (Generalized Trace Facility) 22
H
HDB
  exporting data sets to DB2 441
  housekeeping 439–440
  list 437
  maintenance 437
  overview 416
  primary menu 418
  register 419
  register KSDS 439
  templates 419
  HDB LOAD 431
  HDB Manager 425
  HDB Report panel 427
  HDB Reporting panel 425
  HFS 270
  HFS, jvm.properties file location 270
  HIGHOFFLOAD 225
  Historical Database (HDB) 23
  list HDB 25
  summary HDB 25
  Historical Database Menu 441
  historical performance database xiii, 470
  home interface 131
  HP-UX 242
  HTTP header 262
  HTTP protocol 258
  HTTPNSSL TCPIPSERVICE 263
  HTTPS 243
I
I SMF command 288, 302
IBM CICS Transaction Gateway (CTG) 242
IBM Fault Analyzer for z/OS and OS/390 5
IBM File Manager for z/OS and OS/390 5
IBM Session Manager for z/OS 3–4, 9
IBM VTAM 9
IBM WebSphere Application Server 242
IDCAMS VERIFY 439
IFASMFDP 388
IIOP inbound requests 131
IIOP listener regions 130
IMS 6
MRO 5, 33
MROLRM SIT parameter 252
msglength 206
multithread enterprise bean 300
MVGT transaction 208
MVPT transaction 213
MVS ID (CICS) 101
MVS Workload Manager 24
MVS Workload Manager (WLM) service 61

N
NEW command 36, 48, 56
New Report Form 48
nonshared resources (NSR) 140
non-threadsafe environment 174
non-volatile coupling facility 224
NSR (nonshared resources) 140
nummsgs 206

O
Object List
  creation 146
  EJB example 308
  excluding CICS system transactions 364
  maintaining 56
  VSAM example 146
Open Transaction Environment (OTE) 171, 173
OS/390 242
Overall CICS System Usage 73

P
padchar 206
Parallel Sysplex 130
performance class clock field 444
performance class data 11, 61–63
  summarized by transaction and program 184
performance class record 35, 65, 67, 70, 73, 78, 90, 207
performance criteria 364
performance data 25, 105
performance data record 10
Performance extracts 99
performance fields 359
Performance Graph report 63, 97
Performance List Extended report 61, 67
  default format 68
  export to extract data set 69
  tailoring 68
  VSAM example 153
Performance List report 61, 65
  Application Naming example 312
  CICS Transaction Gateway example 251
  CICS Web Support example 264
  DBCTL example 66
  default format 65
  EJB example 294
  export to extract data set 67
  Java example 275
  tailoring 66
  VSAM example 152
performance objectives 4, 142, 158, 387
performance record 359
  making a subset 356
Performance reports 65
Performance select statement 148, 154
Performance selection criteria 51, 57
Performance Summary report 61, 70
  CICS Transaction Gateway example 250, 255
    default format 70
  EJB example 294
    export to extract data set 72
  Java example 276
    tailoring 70
  VSAM example 145
Performance Totals report 61, 72, 362
persistence 206
Personal Communications 150
pool threads 173
Primary Option Menu 29
PROGRAM definition 269
programs 7
PROTECTNUM parameter 173
PTF 12
UQ68396 26
UQ68398 26
UQ68400 26
UQ70905 26
UQ70908 26
UQ70913 26

Q
qname 206
QR TCB 174, 434
quasi-re-entrant TCB 174
QUASIRENT 417

R
Recap report 101
Record Level Sharing (RLS) 140
Record Selection extract 63, 104
Record Selection report 271
record type 88 16
Redbooks Web site 456
  Contact us xv
Report Form
  data set 44
  field categories 48
  list 45
  sample 45
  TDSUM 46
  types 44
  using in your report 49
report format default 65
Report Set 60
  creation 147
  Global Options 38
REQUESTMODEL definition 287
Resource Management Facility 22
Resource Manager Interface (RMI) 417
Resource Utilization statistics 73
restricting report output size 373
RESUSAGE(FILESUM) command 64
RESUSAGE(TRANLIST) command 64
RFIND command 53
RLS (Record Level Sharing) 140
RMI MQ field 221
RMI parameter 207
RMIMQM field 216
RO CPU Time 359
RUN command 54, 191
S
SCEERUN library 320
SCEESAMP library 320
SCS-type printers 8
SDFHLOAD 35
selection criteria 52
Session Manager for z/OS 9
SETTINGS command 27
sharable application classpath 270
single thread enterprise bean 287
SIT parameters
AKPFREQ 226, 234
LGDFINT 228
MAXVMTCBS 270, 277
MAXOPENTCBS 176, 194, 199
MN 10, 164
MNEXC 10
MNPER 10, 20
MNRES 10, 163
MROFSE 251
MROLRM 251
RUWAPOOL 320, 336
STATRCD 19
TCBLIMIT 187
TCP=YES 363
SMF (system management facility) 5, 25
SMF 101 DB2 accounting records 63
SMF 101 record 14
SMF 110 CMF performance records 63
SMF 110 record 10
SMF 116 record 15
SMF 116 WebSphere MQ accounting records 63
SMF command I SMF 28
SMF dump program (IFASMFDP) 28
SMF records
CICS Monitoring Facility (110) 5
DB2 Accounting (101) 5, 14
SMF 101 DB2 accounting records 63
SMF 110 CMF performance records 63
SMF 116 WebSphere MQ accounting records 63
System Logger (88) 5, 16, 226
Socket Listener 258
socket TCB 266
SOCKETCLOSE 258
SOCKETCLOSE parameter 257
SPUFI 449
SQL statements

GRANT 172
REVOKE 172
SELECT with FETCH FIRST 10 ROWS ONLY option 382
SQLJ 269, 287
SQLJ interface 135
SQLJ support in CICS 268, 287
SQLJ workload 302
SSL 243
Staging Data Set Threshold Hit condition 232
staking-data-set-full event 227
stateful session beans 269
stateless CORBA objects 268
stateless session beans 269
STATISTICS 19
structure alter event 95
Subsystem report 62, 89
SUMEXC command 64
SUMMARY command 64
Summary Report Form 145
Sun Solaris 242
suspend components 77
SYNCONRETURN option 251
syncpoint 206, 251
SYSDEFLT 444
SYSIBM.SYSTABLES 449
System Logger 6
List report 238
Logstream Summary report 232
report 94
system management facility (SMF) 5, 25
System reports 62, 94
system resources manager (SRM) 22

Take-up command 188
Take-Up from SMF 249
update information 159
Take-up function 208
task-related user exit 173
TCB
open (L8) 175
quasi-reentrant 266
socket 266
thread 173
tcb limit parameter 176, 194, 197
TCP/IP 131
TCP/IP interfaces 253
TCP62 243
TCPSERVICE definition 253, 257–258
Teleprocessing Network Simulator (TPNS) 130,
140–141, 386
temporary storage 78
queues 7
Temporary Storage Usage Summary report 62, 87
terminals 7
THREADLIMIT 198, 203
THREADLIMIT parameter 187, 193–194
THREADSAFE 417
threadsafe environment 175
time zone    38
Tivoli Decision Support   19
Tivoli Decision Support for OS/390   22
TOTAL command   64
TRANGROUP command   64
transaction   7
transaction classes   7
transaction execution phases
  begin-to-end   379
  execution   379
Transaction Group option   260
Transaction Group report   35, 80, 260
Transaction Rate Graph report   63, 97–98
transaction resource class data   12, 26, 62
transaction resource data record   11
Transaction Resource Monitoring   14, 165
Transaction Resource Monitoring support   163
Transaction Resource Usage List report   62, 168
Transaction Resource Usage report   62, 85, 88
Transaction Response Time Graph report   63, 98–99
transaction routing   130
Transient Data (D)   78
transient data queue   7
  CESE   321
TSPUTAux column   217
TXSeries CICS Servers   243
TYPE=INITIAL macro   207

U
UOW   186
User Field statistics   73
user fields   13, 136, 319
user journals   224
user-replaceable modules (URM)
  DFHWBADX   259
  program autoinstall   338

V
VERIFY command   441
view sets
  CICS dispatcher   6
  CICS region   6
  CICS storage (DSA and subpool usage)   6
  DB2 connection   6
  enqueue pools   6
  global task related user exits   7
  JVM pool   7
  loader   6
  recovery manager, including UOW analysis   6
  temporary storage   6
  transaction manager   6
  transient data   6
  volatile coupling facility   224
VSAM   386
  CICS-VSAM interface   xiii, 470
  KSDS   416
  Local Shared Resource   130
  nonshared resource   140
  Record Level Sharing   130, 140
VSE/ESA V1   242
VTAM Generic Resource   130

W
Wait Analysis report   61, 75, 398
WAITANAL command   64
WebSphere Accounting   206
WebSphere Application Server
  Application Assembly Tool   245, 252
  connectivity with CICS   243
WebSphere MQ   6, 24, 205–206
  accounting data   15
  API calls MQGET and MQPUT   206
  CICS PA reports overview   206
  MQFI START TRACE   207
  queue manager   206
  SMF accounting records   205
  transactions
    MVGT,nummsgs,datatype,syncpoint,qname   206
    MVPT,nummsgs,padchar,msglength,persistence,qname   206
WebSphere MQ reports   62, 91, 134, 211
  Class 1 List and Summary reports   134
  Class 3 List and Summary reports   134
  WebSphere Studio Application Developer   268
  Windows 2000   241
  Windows NT   242
  wizard icon   115
Workload Activity Report   61, 82–83
  Workload Manager Activity List report   378
  Workload Manager Activity Summary by Service Class   381
  Workload Manager Activity Summary report   380
Workload Activity report   35
  Workload Manager Activity List report   378
  Workload Manager Activity Summary by Service Class   381
  Workload Manager Activity Summary report   380

X
XMINOUT   164
Xresettable parameter   270

Z
z/OS   241
  CICS Transaction Gateway   131
  Generalized Trace Facility   22
  Lightweight Directory Access Protocol   131
  Resource Measurement Facility   61
  Resource Recovery Services   255
  System Logger   224
  System Management Facility   5
zSeries tools
  Application Development   5
  CICS tools   5
  data management   5
  system management   5
CICS Performance Analyzer

Gain insight to the IBM WebSphere MQ reporting capabilities

Explore enhanced CICS Monitoring Facility (CMF) resource class reporting

Learn how to work with historical performance data

This IBM Redbooks publication targets CICS Transaction Server V1.3 and V2.2 customers who plan to implement IBM CICS Performance Analyzer. With this tool, you can produce a wide range of reports and extracts to help you tune and manage CICS systems.

The first part of this book begins with an overview of CICS-provided tools and utilities that help you gather and analyze performance data. It introduces the CICS Performance Analyzer product and its various report generating options. It also shows you how to import the extracted performance data into spreadsheets for further analysis.

The second part of this book takes you through a series of scenarios that cover major CICS components and interfaces. These include CICS-VSAM interface, CICS-DB2 Attachment Facility, CICS use of the MVS System Logger, Java applications in CICS, and others. For each scenario, you see how you can extract the relevant performance data using CICS Performance Analyzer. You can then use this data to improve the overall system performance or to compare different execution options at run time.

This Redbooks publication explores the new functionality of CICS PA Release 1.3, including IBM WebSphere MQ and how CICS PA now handles System Management Facility (SMF) 116 records. It looks at the new CICS Monitoring Facility (CMF) reports such as the Wait Analysis and Temporary Storage Usage reports. It also explains the Historical Database (HDB) facility for maintaining a history of CMF performance data for longer term reporting or exporting to DB2.

For more information: ibm.com/redbooks

SG24-6063-00 ISBN 0738499013